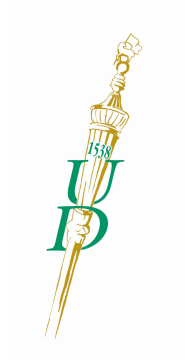


TDK dolgozat

Kreith Balázs



DEBRECENI EGYETEM ORVOS- ÉS EGÉSZSÉGTUDOMÁNYI CENTRUM
BIOFIZIKAI ÉS SEJTBOLÓGIAI INTÉZET

Multiple- τ hardware korrelátor implementálása FPGA kártyára FCS és FCCS mérésekhez



Témavezetők:

Dr. Vámosi György (tudományos főmunkatárs)

Mocsár Gábor (PhD hallgató)

Készítette:

Kreith Balázs

Debrecen, 2011

Bírálatra való beküldéséhez hozzájárulok.

Kivonat

Az elmúlt évtizedekben a fluoreszcencia korrelációs és keresztkorrelációs spektroszkópia (FCS, FCCS) széleskörűen elterjedt biofizikai módszerré vált molekulák mozgásának és kölcsönhatásainak tanulmányozására. TDK dolgozatomban egy multiple- τ hardware korrelátort írok le, amelyet FCS és FCCS mérések auto- és keresztkorrelációs számításaihoz fejlesztettünk ki. Ütemező eljárásunk alkalmazása révén a korrelátor egyetlen lineáris korrelátor blokk segítségével számítja ki két detektor jelének auto- és keresztkorrelációs függvényeit 400 ns időfelbontással. A korrelátort LabVIEW környezetben implementáltuk egy National Instruments gyártmányú FPGA (field programmable gate array) kártyán.

Abstract

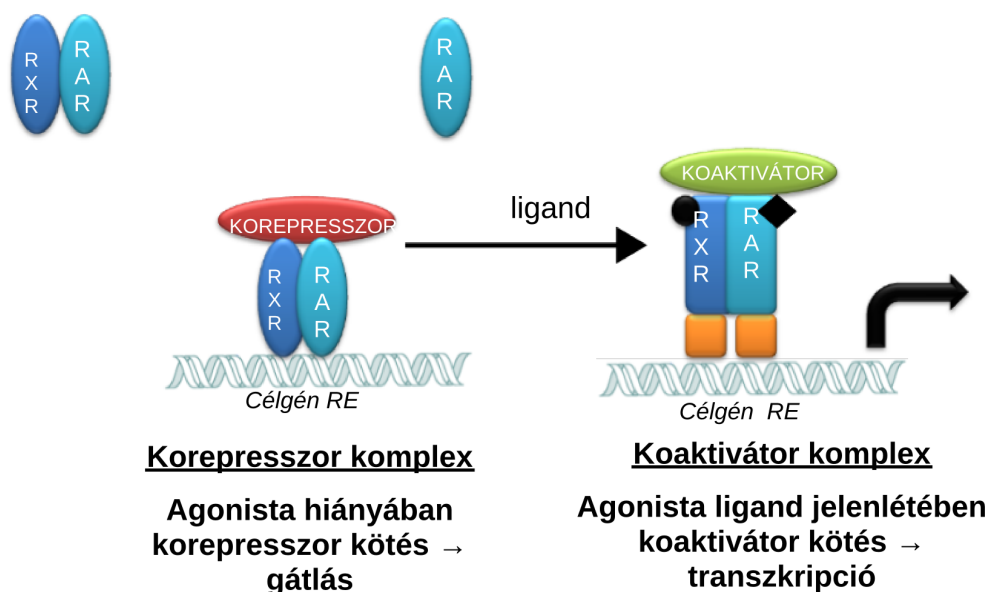
In the last decades fluorescence correlation and crosscorrelation spectroscopy (FCS, FCCS) have become widely used biophysical techniques to determine diffusion properties and interactions of fluorescing molecules in living cells. Here a novel multiple- τ hardware correlator design is introduced for computing auto- and crosscorrelation functions from FCS and FCCS measurements. Our design applies a custom scheduler algorithm, which allows for calculating the auto- and crosscorrelation functions of the signals from two detectors on a single linear correlator block with 400 ns time resolution. The correlators are implemented in LabVIEW environment on a National Instruments FPGA (field programmable gate array) card.

Tartalomjegyzék

1. Bevezetés	1
1.1. A dolgozatban alkalmazott jelölések	3
2. A fluoreszcencia korrelációs- és keresztkorrelációs spektroszkópia	4
2.1. Autokorreláció becslése digitális korrelátorokkal FCS mérésekhez	7
2.2. A multiple- τ korrelátorok	9
3. Multiple-τ korrelátor tervezése	12
3.1. Optimalizálás	12
3.2. Ütemezés	13
3.3. Korrelátorblokk végrehajtási egység	14
3.4. Correlation Processing Element (CorrPE)	15
3.4.1. CorrPE-1	16
3.4.2. CorrPE-2X	17
3.5. A kiszámított korrelációs értékek kiolvasása és normalizálása	19
4. Multiple-τ korrelátor implementálása	21
4.1. FPGA használata digitális korrelátorok megvalósításához	21
4.2. CorrPE-1 és CorrPE-2X hardware korrelátorok	22
4.3. CorrPE-App	24
5. Eredmények	25
6. Összefoglalás	27
6.1. Saját munka leírása	28
6.2. Köszönetnyilvánítás	28
A. Függelékek	29
A.1. A (3.2)-es egyenlethez kapcsolódó állítások és bizonyítások	29
A.2. A (3.3)-as egyenlethez kapcsolódó állítások és bizonyítások	30

1. Bevezetés

A molekuláris kölcsönhatások a sejtbiológiai folyamatokban (pl: fehérje-fehérje interakciók) kulcsszerepet töltenek be. Ezek a folyamatok jól nyomonkövethetőek a molekulák mobilitásának, valamint együttmozgásának vizsgálatával. Erre a célra alkalmas módszer a fluoreszcens korrelációs és keresztkorrelációs spektroszkópia (FCS, FCCS), amely az elmúlt évtizedben széles körben használt biofizikai technikává vált molekulák mozgásának tanulmányozására oldatokban vagy élő sejtekben. Az FCS mérések segítségével meghatározható a vizsgált molekula diffúziós ideje és koncentrációja, az FCCS mérések kiértékelésével pedig következtethetünk a megfigyelt molekulák közötti kapcsolatokra.



Nagy et al, TRENDS in Biochemical Sciences 2004. alapján

1.1. ábra. Az ábrán a DNS-hez dimer formájában kötődő, a gének működését szabályozó fehérjék láthatóak. Ezek aktiváló vagy gátló komplexben helyezkednek el, vagy pedig szabadon diffundálnak a sejtmagban. A szabad magreceptor mozgása gyors, a DNS-hez kötött receptoré lassú. A kötődés kimutatására így alkalmas a mobilitás mérése. A működés megértéséhez szükség van arra, hogy meghatározhassuk az egymással és a DNS-sel kölcsönható magreceptorok mobilitását külön-külön, illetve kölcsönhatásuk esetén együttmozgásukat is. Ezeket FCS és FCCS mérésekkel lehet meghatározni.

A Debreceni Egyetem Orvos- és Egészségtudományi Centrum Biofizikai és Sejtbiológiai Intézetének Sejtbiológiai és Jelátviteli Kutatócsoportjában jelenleg a kétféle mérést csak egymás után lehet elvégezni, azonban a biológiai rendszerek dinamikusak és mikroszkopikus szinten pillanatról pillanatra változnak, így a pontosabb model felállításához az FCS és FCCS egyidejű mérésére van szükség. Célunk egy olyan műszer kifejlesztése volt, mely szimultán képes FCS és FCCS mérésekhez szükséges auto- és keresztkorrelációs függvények kiszámítására.

Az FCS és FCCS mérésekhez szükséges auto- és keresztkorrelációs függvények kiszámítása a méréssel egyidőben történik 400ns és ~ 1 s között. A célhardware a korrelációs függvényeket a multiple- τ koncepció alapján számolja ki, amelyhez fejlesztett és eddig nem publikált ütemezési eljárás a felhasznált erőforrásokat minimalizálja.

1.1. A dolgozatban alkalmazott jelölések

Jelölés	Leírás
K	Digitális áramkörökben implementálásra került korrelációs csatonák száma
L	Korrelátorblokk korrelációs csatornáinak a száma
m	A mintavételezési idő korrelátorblokkok közötti növekményének az értéke
S	Korrelátorblokkok száma
τ	Időállandó
$\tau_{\min}$	A minimális mintavételezési idő állandója.
$\tau_{\max}$	Az elérni kívánt maximális felbontás időállandója.
$\Delta\tau_{lin}$	Korrelátorblokk egy végrehajtásához szükséges időállandó.
s	Tetszőleges korrelátorblokk azonosítója ($s \in [1..S]$).
l	Tetszőleges korrelációs csatorna azonosítója egy korrelátorblokkon belül ($l \in [1..L]$).
τ_k	Az időállandók halmazának k -adik értéke.
$\tau_{s,l}$	Az s -edik korrelátorblokk l korrelációs csatornájához tartozó időállandó értéke.
I_n	A detektált fotonok száma egy az n -edik mintavételi periódusban ahol a periódusok egymást követő azonos hosszúságú időintervallumok.
$I_s^{(u)}$	Egy s -edik korrelátorblokk aktuális "undelayed" periódusában mért fotonoknak a száma.
$I_s^{(u)*}$	Egy s -edik korrelátorblokk az aktuálist egyel megelőző periódusban mért fotonoknak a száma.
$I_s^{(u,x)}$	Az $I_s^{(u)}$ értéke az x bemeneti jelhez az s -edik korrelátorblokkban.
$I_s^{(u,x)*}$	Az $I_s^{(u)*}$ értéke az x bemeneti jelhez az s -edik korrelátorblokkban.
$I_{s,k}^{(d)}$	Egy s -edik korrelátorblokk I_{n-k} értéke.
$I_{s,k}^{(d,x)}$	Az $I_{s,k}^{(d)}$ értéke az x bemeneti jelhez az s -edik korrelátorblokkban.
G_n	Az n -edik korrelációs csatornában kiszámolt érték.
$G_{s,n}$	Az s -edik korrelátorblokk n -edik korrelációs csatornájához tartozó érték.
$G_{s,n}^{(x,y)}$	Az x és y bemenet között számolt korrelációs csatorna értéke az s -edik blokk n -edik korrelációs csatornájában.
M_n	Egy n -edik monitorcsatorna értéke az aktuális peroidusban.
$M_{s,n}$	Egy s -edik korrelátorblokk n -edik monitorcsatornájának értéke az aktuális periódusban.
$M_{s,n}^{(x)}$	Az x bemenethez tartozó $M_{s,n}$ érték.

2. A fluoreszcencia korrelációs- és keresztkorrelációs spektroszkópia

A fluoreszcencia korrelációs spektroszkópia (FCS) egy hatékony biofizikai módszer molekuláris mozgások tanulmányozására oldatokban vagy élő sejtekben. A módszer alkalmazása során fókuszált lézernyaláb effektív térfogatán (kb. 0,1 femtoliter) keresztülhaladó fluoreszkáló részecskék emittált fotonjait detektáljuk, általában lavina fotodiódával (APD). A fluoreszcencia intenzitás fluktuációja a fluoreszkáló részecskék mozgékonyaságától, koncentrációjától, illetve minden olyan fotofizikai folyamattól is függ, amely hozzájárul a molekula kvantumhatásfokának megváltozásához, például a triplet állapot vagy egy nem emittáló „sötét állapot” kialakulása ([1]). A molekulák mozgásának tulajdonságai (diffúziós állandó, koncentráció az adott térfogatban ([5]) a fluktuációk ingadozásából autokorrelációs függvény ($g(\tau)$, ACF) segítségével meghatározhatóak.

$$g(\tau) = \frac{\langle \delta I(t) \cdot \delta I(t + \tau) \rangle}{\langle I(t) \rangle^2}, \quad \delta I(t) = I(t) - \langle I(t) \rangle \quad (2.1)$$

Ahol $\delta I(t)$ egy adott t időpillanatban a fluoreszcencia fluktuáció értéke, $\langle I(t) \rangle$ az átlagos fluoreszcencia intenzitás. Amennyiben a részecskék diffúziós mechanizmusa ismert, úgy az autokorrelációs görbe analitikus alakja is meghatározható. Pl.: a 3D-ban szabadon diffundáló egykomponensű fluoreszcens részecskék autokorrelációs görbáját a következő képlettel lehet megadni, ha a detektálás hatásfokának térbeli eloszlása 3D Gauss-eloszlást követ ([6]).

$$g(\tau) = \frac{1}{N} \cdot \left(1 + \frac{\tau}{\tau_d}\right)^{-1} \cdot \left(1 + \frac{1}{S^2} \cdot \frac{\tau}{\tau_d}\right)^{-1/2} \quad (2.2)$$

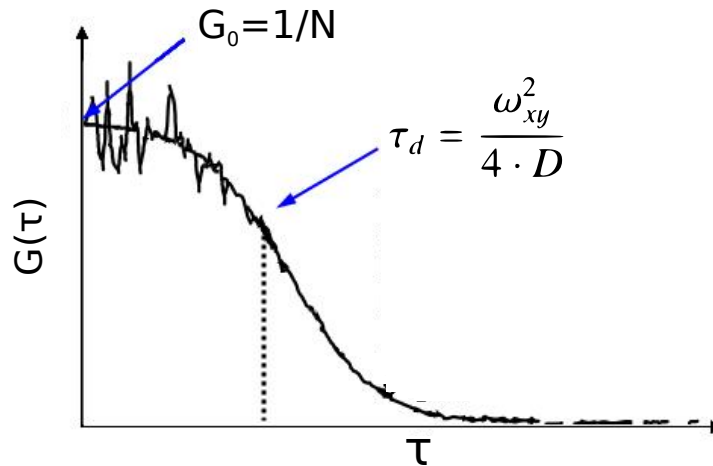
Ahol N a detektálási térfogatban jelenlévő fluoreszcens molekulák átlagos száma, S az ellipszoid alakúnak feltételezett detektálási térfogat axiális és laterális tengelyének hányadosa, τ_d a komponens diffúziós ideje, az az átlagos időtartam, ami alatt a molekula áthalad a detektált térfogaton.

A molekulák diffúziós állandója (mozgékonyysága) a diffúziós időből és a detektálási térfogat ω_{xy} laterális sugarából a következőképpen határozható meg:

$$\tau_d = \frac{\omega_{xy}^2}{4 \cdot D} \quad (2.3)$$

Az ACF-ből nyerhető másik alapvető fontosságú paraméter a lokális koncentráció: a függvény amplitudója, $\tau = 0$ -ra extrapolált értéke $G_0 = G(0) = 1/N$, azaz a részecskeszám reciproka.

Az autokorrelációs függvény szemléletes jelentése diffundáló molekulák esetén a következő: $g(\tau)$ értéke arányos annak valószínűségével (ha az amplitudót 1-re normáljuk), hogy ha egy molekula a $t = 0$ időpillanatban a detektálási térfogatban volt, akkor a molekula a $t + \tau$ időpillanatban is a detektálási térfogatban lesz.



2.1. ábra. Diffundáló molekula autokorrelációs függvénye . Az ábrán látható a diffundáló molekula intenzitás-fluktuációira számolt mért és megillesztett autokorrelációs görbe.

Több diffundáló komponens esetén az autokorrelációs görbe képlete a következő:

$$g_{diff}(\tau) = \sum_i w_i \left(1 + \frac{\tau}{\tau_{d,i}}\right)^{-1} \cdot \left(1 + \frac{1}{S^2} \cdot \frac{\tau}{\tau_{d,i}}\right)^{-1/2} \quad (2.4)$$

Ahol w_i az i -edik komponens súlya (ami a koncentrációjától és relatív fényességétől függ), $\tau_{d,i}$, az i -edik komponens diffúziós ideje.

Amennyiben a fluktuáció ingadozásához több folyamat is közrejátszik, a konkrét autokorrelációs görbéhez az egyes folyamatok autokorrelációja is hozzájárul. Ha a fluoreszcens részecskék triplet állapotát is figyelembe vesszük, akkor az ACF a következő képlettel adható meg [13]:

$$g(\tau) = \frac{1}{N} \cdot g_{triplet}(\tau) \cdot g_{diff}(\tau), \quad g_{triplet}(\tau) = \left(\frac{1 - T + T e^{-\tau/\tau_{tr}}}{1 - T} \right) \quad (2.5)$$

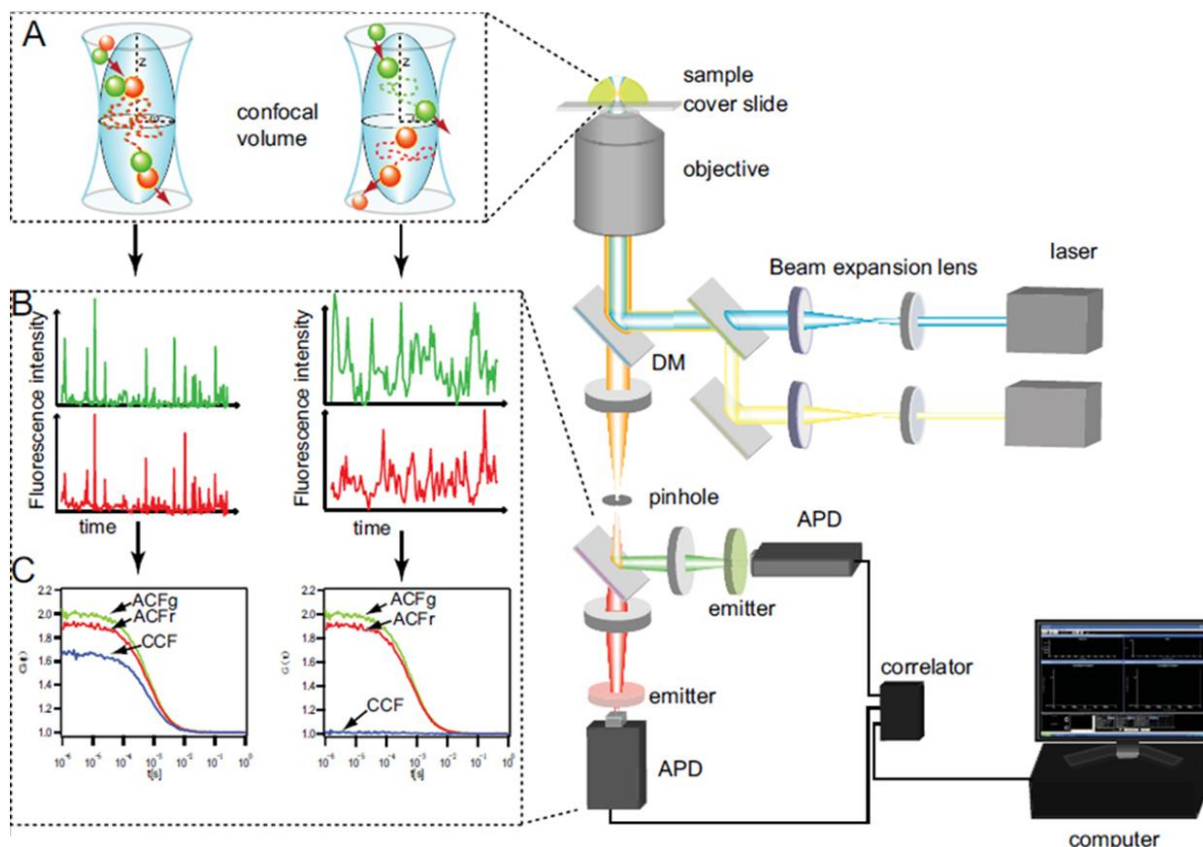
Ahol T a triplet állapotban lévő molekulák hányada τ_{tr} a triplet (foszforeszcencia) élettartam. Az FCS mérések jól használhatóak a különféle molekulák egyedi mozgásainak tanulmányozására élő sejtekben ([12, 11]), és alkalmasak pl. két molekula kölcsönhatásának kimutatására, ha az a vizsgált molekulák diffúziós idejének jelentős megváltozásával jár. De nem elég érzékeny pl. azonos molekulatömegű molekulák asszociációjának kimutatásához. Két molekula mozgására vonatkozó paraméterek autokorrelációval való meghatározásából nem tudunk következtetni arra, hogy ez a két molekula együtt haladt-e át a detektálási térfogaton. Ahhoz, hogy két molekula együttes mozgásáról kapjunk információt, a detektált fluoreszcencia fluktuációk között számolunk keresztkorrelációt az alábbi definíció alapján ([7]):

$$g(\tau) = \frac{\langle \delta I_a(t) \cdot \delta I_b(t + \tau) \rangle}{\langle I_a(t) \rangle \cdot \langle I_b(t) \rangle} \quad (2.6)$$

Ahol „a” és „b” jelölik azokat a detektorokat amelyek mért intenzitása között számolunk korrelációt. Amennyiben az „a”-t jelölő detektor megegyezik a „b”-t jelölő detektorral, úgy autokorrelációt, egyébként keresztkorrelációt számolunk. A fluoreszcencia keresztkorrelációs spektroszkópia (FCCS) azon alapszik, hogy ugyanazon detektálási térfogatból származó, különböző fény-hullámhosszon fluoreszkáló molekulák fluoreszcens fluktuációja között számolt keresztkorrelációból következtethetünk a vizsgált molekulák közötti interakciókra. Ha feltesszük, hogy a két különböző detektor által detektált fotonok emissziós spektruma között nincs átfedés, valamint hogy a két vizsgált molekula kvantumhatásfoka megegyezik akkor a két molekulára számolt auto és keresztkorrelációs függvény az átlagos koncentráció és az adott molekula diffúziós idejét figyelembevéve a következőképpen definiálható:

$$G_{ab}(0) \propto \frac{c_{ab}}{(c_{a,tot} \cdot c_{b,tot})} \quad (2.7)$$

Ahol c_{ab} a komplex, $c_{a,tot}$ és $c_{b,tot}$ pedig az a és b molekula teljes koncentrációja. A következő ábra egy konfokális mikroszkópon beállított FCCS mérést szemléltet.



2.2. ábra. Egy FCS, FCCS mérés folyamatának szemléltetése. © Biophysical Fluorescence Laboratory, National University of Singapore

A lézerrel megvilágított konfokális térfogaton keresztülhaladva (A) a fluoreszkáló részecskék fotonokat emittálnak, melyek intenzitását (B) APD detektorok segítségével mérjük. Az intenzitásokhoz számolt auto- és keresztkorreláció kiértékelésével (C) a vizsgált molekulák kinetikai tulajdonságai meghatározhatóak. ACFg a zöld, ACFr a vörös csatorna autokorrelációs függvényét, CCF a két csatorna keresztkorrelációs függvényét jelöli

A méréssel szimultán korreláció elkészítéséhez digitális korrelátorokat használunk. A digitális korrelátorok a korrelációs függvényt általában a $0.1\mu s - 1s$ közötti időintervallumon határozzuk meg. A számítás elvégzéséhez szükséges elméleti és gyakorlati háttérrel a következő fejezetekben ismertetem.

2.1. Autokorreláció becslése digitális korrelátorokkal FCS mérésekhez

A digitális korrelátorok célja, hogy a méréssel szimultán meghatározzák a korrelációs függvény értékeit a $\{\tau_k\}$ halmaz által megadott időpillanatokban. A korrelátorok adott mintavételezési frekvenciával (f_{clk}) rendelkeznek, amely meghatározza az autokorrelációs görbe legkisebb

időbeli felbontását ($\tau_{\min} = 1/f_{clk}$). A detektált fotonok száma az egymást követő $\tau_{\min}$ hosszúságú időintervallumokban a következő képlettel írható le:

$$I_n = \int_0^{\tau_{\min}} I(n \cdot \tau_{\min} + t) dt, \quad n = 0, 1, \dots, T \quad (2.8)$$

Ahol T a mérés teljes időtartama $\tau_{\min}$ egységekben, n az n -edik mintavételi periódust jelöli. A (2.1)-es egyenletben definiált képlet a normalizált elméleti autokorrelációs függvény képlete, ahol τ a $[0, \infty]$ közti időintervallumon vehet fel értékeket. A hardware korrelátorok az autokorrelációs görbe becslésére a következő, ún. szimmetrikus normalizálást alkalmazzák:

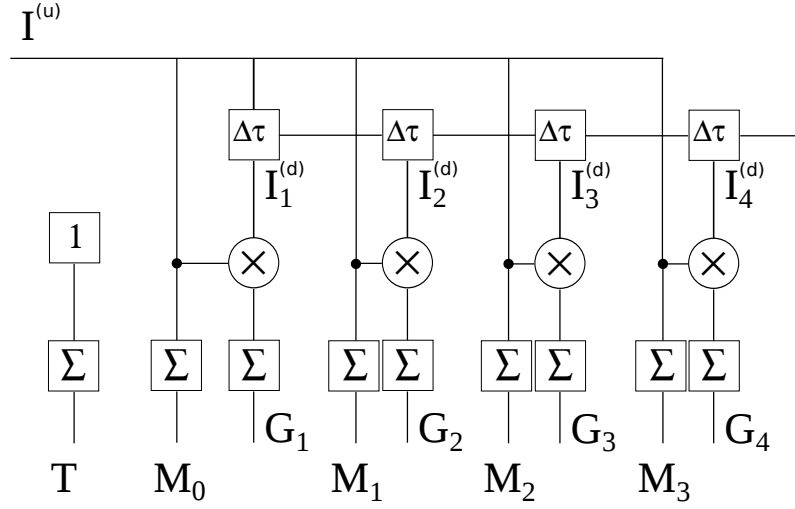
$$\hat{g}_{symm}(\tau_k) = \frac{\overbrace{\frac{1}{T-\tau_k} \cdot \sum_{n=\tau_k}^T I_n \cdot I_{n-\tau_k}}^{G_{\tau_k}}}{\underbrace{\frac{1}{T} \cdot \sum_{n=0}^{T-1} I_n}_{M_0} \cdot \underbrace{\frac{1}{T-\tau_k} \cdot \sum_{n=\tau_k}^T I_{n-\tau_k}}_{M_{\tau_k}}} \quad (2.9)$$

Ahol τ_k egy $\tau_{\min}$ egységekben megadott időpont, amelyen az autokorrelációs görbét kiszámítjuk. Azon időpontok számát, ahol a korrelációs függvényt kiszámítjuk, K -val jelöljük ($K = |\tau_k|$). A korrelátor megvalósításához felhasznált erőforrások szempontjából K értéke egy fontos paraméter. A meghatározandó momentumok közötti időközleltetést $\Delta\tau_k$ -val jelöljük, ahol $\Delta\tau_k = \tau_k - \tau_{k-1}$.

A mérés során az n . periódusban az aktuális intenzitás értéke I_n ami az ún. „undelayed” intenzitás ($I^{(u)} = I_n$). Az I_{n-k} az aktuális periódushoz képesti τ_k időegységgel korábbi „delayed” intenzitás értéke ($I_k^{(d)} = I_{n-k}$).

Az implementáció során az egyenletben megjelölt G_{τ_k} , M_{τ_k} és M_0 értékek kiszámítása és tárolása a célhardware feladata. Ennek elvégzéséhez szükség van az intenzitási értékekre (I_n). A digitális eszközökön a G_{τ_k} érték kiszámítása egy ún. korrelációs csatornában, az M_{τ_k} és M_0 értékek kiszámítása egy ún. monitor csatornában hajtodízik végre. Az I_n értékeit az ún. intenzitási csatorna tartalmazza. Az implementált korrelációs csatornák száma K .

Az ún. lineáris korrelátorokban a meghatározandó momentumok értékei egyenletes időközökkel rendelkeznek a minimális és a maximális időbeli felbontás között ($\Delta\tau_1 = \Delta\tau_2 = \dots = \Delta\tau_n = \tau_{\min}$). Egy lineáris korrelátor sematikus vázát és a benne található korrelációs csatornákat szemlélteti a következő ábra.



2.3. ábra. Lineáris korrelátor sematikus rajza.

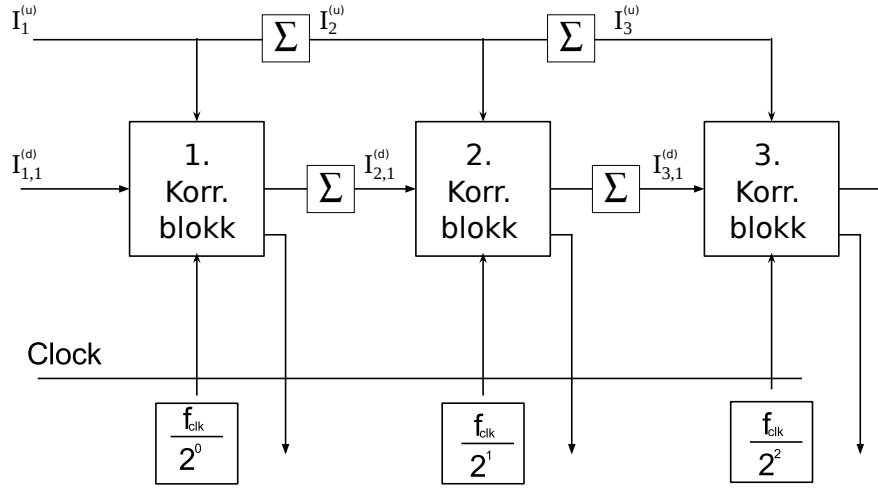
A lineáris korrelátorok esetében a megvalósításhoz szükséges korrelációs csatornák száma (L) megegyezik a kiszámolandó momentumok számával (K). Egy FCS mérés során az ACF-t legalább $1\mu s$ és $1s$ közötti időintervallumban kell meghatározni, ami 10^6 korrelációs csatorna implementálását jelentené, ezért a mérések kiértékeléséhez használt hardware korrelátorok egy kvázi-logaritmikus időskálát alkalmaznak, és az ún. multiple- τ koncepciót valósítják meg.

2.2. A multiple- τ korrelátorok

A multiple- τ koncepció szerinti megvalósításban lineáris korrelátorok láncszerűen vannak egymáshoz kapcsolva ([8, 9]). Az összekapcsolt lineáris korrelátorok (a következőkben korrelátorblokkok) száma S . A legtöbb multiple- τ koncepció szerint megvalósított korrelátorban a korrelátorblokkok azonos (L) számú korrelációs csatornát tartalmaznak. Egy korrelátorblokkon belül a korrelációs csatornák közötti időkésltetés egyenlő a korrelátorblokk mintavételezési idejével, ami blokkról blokkra növekszik. Egy blokk mintavételezési ideje egy öt megelőző blokk mintavételezési idejének m -szerese.

$$\Delta\tau_s = \begin{cases} \tau_{\min} & \text{ha } s = 1 \\ \Delta\tau_{s-1} \cdot m & \text{egyébként} \end{cases}, s \in [1 \dots S] \quad (2.10)$$

Ahol s ($0 < s < S$) az s . korrelátorblokkot jelenti. Egy korrelátorblokk mintavételezési frekvenciája ($f_{clk,s}$) megegyezik a $\Delta\tau_s$ érték reciprokával ($f_{clk,s} = 1/\Delta\tau_s$).



2.4. ábra. multiple- τ séma $m = 2$ esetben. Minden lineáris korrelátor mintavételezési frekvenciája az öt megelőző korrelátorblokk frekvenciájának a fele.

A $\Delta\tau_s$ ismeretében az s . korrelátorblokk korrelációs csatornáit által a korrelációs függvényen meghatározott momentumok (τ_k) értékei meghatározhatóak a következőképpen:

$$\tau_{s,l} = \begin{cases} l \cdot \tau_{\min} & \text{ha } s = 1 \\ \tau_{s-1,L} + \Delta\tau_s \cdot l & \text{egyébként} \end{cases}, \quad s \in [1, 2 \dots S], \quad l \in [1, 2, \dots L] \quad (2.11)$$

Ahol $l \in 1 \dots L$ azonosítja a korrelációs csatornát az s -edik korrelátorblokkon belül. A korrelátorblokkok mintavételezési idejeinek növelésével arányosan az adott idő alatt a detektor által érzékelt fotonok száma összeadódik. Az s -edik blokk n -edik mintavételezési periódusában a detektált fotonok számát a következő képlet írja le:

$$I_{s,n} = \sum_{i=0}^{m^{s-1}-1} I_{m^{s-1},n-i} \quad (2.12)$$

Bármely blokkban az aktuális periódus „undelayed” intenzitásának értéke: $I_s^{(u)} = I_{s,n}$, a korrelációs csatornához tartozó „delayed” pedig: $I_{s,l}^{(d)} = I_{s,n-l}$. Amennyiben az APD érzékelt egy foton úgy egy TTL jelet küld a korrelátornak, amelyben ekkor az adott periódusban $I_1^{(u)}$ értéke 1 lesz, ellenkező esetben 0. Az első korrelátorblokk „delayed” intenzitása $I_{1,1}^{(d)}$ megegyezik az előző periódus „undelayed” intenzitás értékével. Minden további blokk első

delayed intenzitásának értékét ($I_{s,1}^{(d)}$) az öt megelőző korrelátorblokk utolsó m intenzitásának az összege határozza meg.

$$I_{s,1}^{(d)} = \sum_{i=1}^m I_{s-1,L+i}^{(d)} \quad (2.13)$$

A normalizált szimmetrikus autokorreláció becslése a multiple- τ séma szerint, az előző jelölések bevezetésével a következő képlettel írható le:

$$\hat{g}_{multi-\tau, symm}(\tau_{s,l}) = \frac{\overbrace{\frac{1}{T_s-l} \cdot \sum_{i=0}^{T_s-1} I_s^{(u)} \cdot I_{s,l}^{(d)}}^{G_{s,l}}}{\underbrace{\frac{1}{T_s} \cdot \sum_{i=0}^{T_s-1} I_s^{(u)}}_{M_{s,0}} \cdot \underbrace{\frac{1}{T_s-l} \cdot \sum_{i=sl}^{T_s} I_{s,l}^{(d)}}_{M_{s,l}}} \quad (2.14)$$

Ahol a $G_{s,l}$ az s -edik korrelátorblokkban egy korrelációs csatorna kiszámításának részösszege, $M_{s,l}$ egy korrelációs csatornához tartozó monitor csatorna, az $I_s^{(u)}$ az aktuális "undelayed" intenzitás értéke az s -edik blokk mintavételezési idejének megfelelően, és s a blokk mintavételezéseinek a száma.

Összességében a multiple- τ séma alkalmazása a láncszerűen egymáshoz kapcsolt lineáris korrelátorblokkokon alapszik. Minden korrelátorblokk végrehajtásának frekvenciája az öt megelőző blokk mintavételezési frekvenciájának az m -ed része, így a végrehajtások közt eltelt idő a blokkok sorszámaival exponenciálisan nő. Minden blokk a korrelációs görbe egy szegmensét számolja ki egy kvázi-logaritmikus skálán. Az implementálandó korrelációs csatornák száma így $K = S \cdot L$, ahol S a kívánt maximális időfelbontás ismeretében ($\tau_{\max}$) meghatározható: $S = \lceil \log_m (\tau_{\max}/\tau_{\min}) \rceil$.

3. Multiple- τ korrelátor tervezése

Az általunk elkészített digitális korrelátorban m értéke 2, L értéke 8, a monitorcsatornák száma blokkonként 1. A következő fejezetekben leírom, a multiple- τ koncepció szerint megvalósított korrelátorokban hogyan lehet az implementált korrelációs csatornák számát lecsökkenteni $S \cdot L$ -ről L -re.

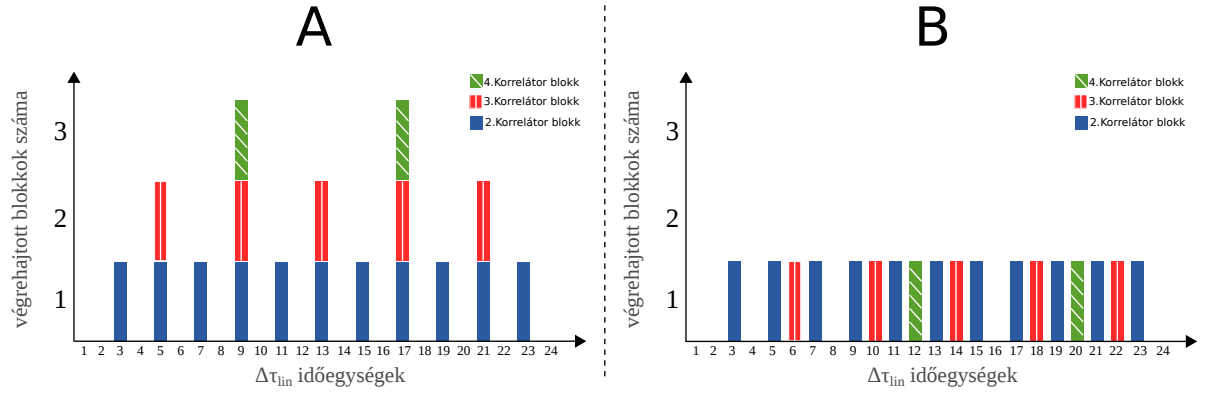
3.1. Optimalizálás

A multiple- τ koncepció alapján ($m = 2$ esetben) az s -edik korrelátorblokk kétszer hajtódik végre, mielőtt az azt követő $(s + 1)$ -edik korrelátorblokk a legutóbbi végrehajtása után újból végrehajtodna. A blokkok végrehajtásának frekvenciája egymás után feleződik. Ha az első $s = 1$ korrelátorblokk n -szer hajtódik végre a mérés során, a második blokk $n/2$ -szer, a 3. $n/4$ -szer, stb. ... akkor az összes korrelátorblokk összes végrehajtása a következő képlettel adható meg (feltéve, hogy $n_1 = 2^S \cdot k$, $k \in \mathbb{N} \setminus \{0\}$):

$$n_{total} = \underbrace{n_1}_{1. \text{ korrelátorblokk}} + \underbrace{n_2}_{2. \text{ korrelátorblokk}} + \dots + \underbrace{n_S}_{S. \text{ korrelátorblokk}} = n_1 \cdot \left(1 + \frac{1}{2} + \dots + \frac{1}{2^{S-1}}\right) \quad (3.1)$$

Ahol n_s az s . korrelátorblokk végrehajtásainak a száma. Az $n_{total} < 2 \cdot n_1$, így az összes blokk együttesen is kevesebbszer hajtódik végre, mint az első blokk végrehajtásainak a kétszerese. Egy blokk végrehajtásához szükséges idő $\Delta\tau_{lin}$.

Az első blokk minden τ_{min} mintevételezésnél végrehajtodik ($\Delta\tau_{lin} = \tau_{min}$), a második blokk $2 \cdot \tau_{min}$ időközönként kap újabb adatot az első bloktól, és a blokk végrehajtása τ_{min} időn belül teljesül. Egy megfelelő ütemezési eljárással $n_2 \cdot \tau_{min}$ időt fel lehet használni a többi blokk végrehajtására.



3.1. ábra. Korrelátorblokkok végrehajtásának száma $\Delta\tau_{lin}$ időegységek függvényében. Az s -edik korrelátorblokk akkor hajtódik végre, ha az őt megelőző $(s - 1)$ -edik blokk kétszer végrehajtott. Az „A” Panel mutatja a párhuzamos végrehajtást és a kihasználatlan időegységeket a blokkok végrehajtásai között. A „B” panelen a blokkok szekvenciálisan vannak végrehajtva egy ütemezési eljárásnak megfelelően.

Amennyiben az első blokk mintavételezési idejét megnöveljük $2^k \cdot \tau_{min}$, $k \in \mathbb{N} \setminus \{0\}$ úgy ezen korrelátorblokk felszabaduló szabad végrehajtási idejeiben az összes többi blokk végrehajtható.

3.2. Ütemezés

Az előző fejezetben leírt optimalizálás megvalósításához az ütemezési eljárás ($m = 2$ esetben) a következő sorozaton alapul:

$$a_c = \begin{cases} 0 & \text{ha } c \text{ páratlan vagy } 0 \\ a_{\frac{c}{2}} + 1 & \text{egyébként} \end{cases}, \quad c = 0, 1, \dots \quad (3.2)$$

A (3.2) egyenletben definiált sorozatban bármely $a_c = k$, $k > 0$ érték előfordulása után az $a_c = k - 1$ érték pontosan kétszer fordul elő a az $a_c = k$ érték újabb megjelenéséig. Ez pedig kielégíti a multiple- τ koncepcióban támasztott feltételt, miszerint bármely $(s - 1)$ -edik korrelátorblokk kétszer hajtódik végre mielőtt az s -edik korrelátorblokk újból végrehajtodna. A megvalósítás során az alábbi, a fenti rekurzív sorozattal ekvivalens iteratív formulát implementáltuk:

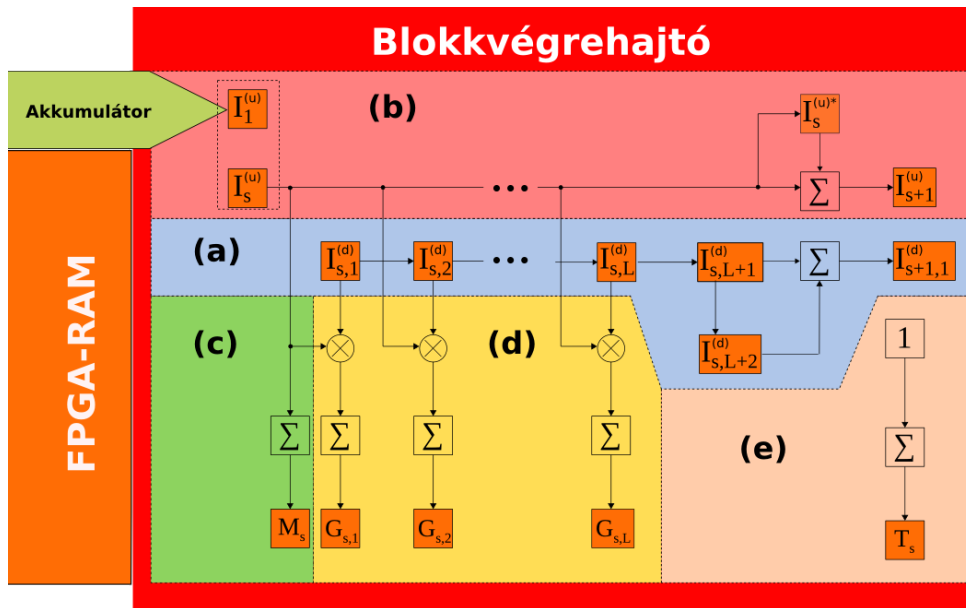
$$a_c = \begin{cases} 0 & \text{ha } c = 0 \\ \log_2(X_c) - 1, & X_c = (c \text{ XOR } c - 1) + 1 \text{ egyébként} \end{cases}, \quad c = 0, 1, \dots \quad (3.3)$$

Ahol a XOR a logikai kizáró vagy műveletet jelenti.

A (3.2) és (3.3) egyenlethez kapcsolódó állítások és azok bizonyításai megtalálhatóak a függelékben. Az ütemező minden egyes c időegységben meghatározza a végrehajtandó korrelátorblokk sorszámát, ami egyenlő $a_c + 1$ -el. A végrehajtáshoz szükség van egy korrelátorblokkokat végrehajtó egységre, amely minden $\tau_{\min}$ hosszúságú ciklusban végrehajtja a megfelelő blokkot.

3.3. Korrelátorblokk végrehajtási egység

A korrelátorblokk végrehajtási egység (továbbiakban blokkvégrehajtó) $\tau_{\min}$ idő alatt teljesíti az egy korrelátorblokkhoz tartozó számítási műveletek összességét. A végrehajtandó blokk azonosítóját, az s értéket paraméterként kapja meg az ütemezőtől, majd a blokk paramétereit ($G_{s,l}, M_{s,0}, I_{s,j}^{(d)}, I_s^{(u)}, I_s^{(u)*}, T_s$ ahol $l \in \{0, 1, \dots, L\}, j \in \{0, 1, \dots, L+2\}$) betölti az FPGA-RAM-ból. Az $(s+1)$ -edik blokk "undelayed" intenzitásának ($I_s^{(u)}$) kiszámításához egy további csatorna ($I_s^{(u)*}$), a következő blokk "delayed" intenzitási érték kiszámításához ($I_{s+1,1}^{(d)}$) további 2 csatorna hozzáadása szükséges, hogy a blokkonként megkétszereződő mintavételezési időre jutó fotonszámokat tároljuk és összegezzük.



3.2. ábra. A blokkvégrehajtó egység felépítése. A betűk jelölik az egyes számítások végrehajtásának helyét, amelyeket a szövegben (lentebb) megjelöltünk. Az FPGA képességeit kihasználva minden művelet párhuzamosan megy végbe

Az ütemező által kiválasztott s -edik blokkban a paraméterek betöltése után a blokkvégrehajtó $\tau_{\min}$ időegység alatt párhuzamosan végrehajtja a következő műveleteket:

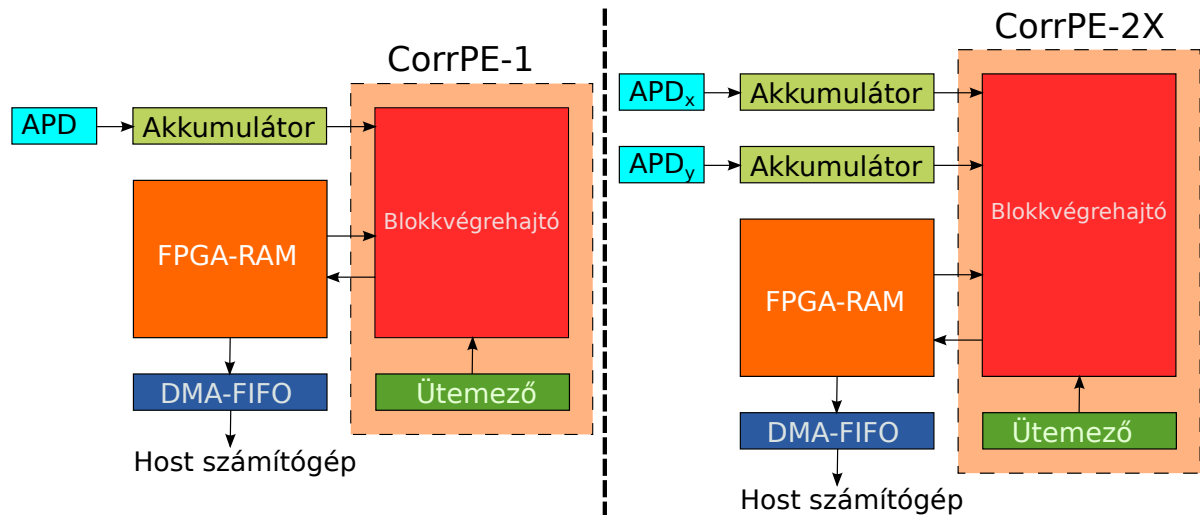
- **(a)** Az intenzitás-csatornában lévő értékeket ($I_{s,j}^{(d)}$) $\Delta\tau_s$ szerint továbbítja a következő módon: Minden $I_{s,j}^{(d)}$ az $I_{s,j-1}^{(d)}$ új értéket kapja az őt megelőző intenzitás-csatornából; A következő blokk késleltetett intenzitásának ($I_{s+1,1}^{(d)}$) kiszámítása pedig az utolsó 2 intenzitás-csatornában lévő értékek összege lesz ($I_{s+1,1}^{(d)} = I_{s,L+1}^{(d)} + I_{s,L+2}^{(d)}$).
- **(b)** A következő blokk késleltetetlen intenzitásának ($I_{s+1}^{(u)}$) kiszámítása ($I_{s+1}^{(u)} = I_s^{(u)} + I_s^{(u)*}$).
- **(c)** M_s értékének megnövelése $I_s^{(u)}$ értékkel.
- **(d)** $G_{s,l}$ értékek inkrementálása $I_s^{(u)} \cdot I_{s,l}^{(d)}$ szorzat értékével.
- **(e)** T_s értékének inkrementálása 1-el.

Az első blokk ($s = 1$) az $I_1^{(u)}$ értékeket közvetlenül egy akkumulátortól kapja. Az akkumulátor $2^k \cdot \tau_{\min}$, $k \in \mathbb{N} \setminus \{0\}$ időegységen keresztül összeadja az APD-től érkező értékeket és a megfelelő időben továbbadja azt az első blokknak. Az $I_{1,1}^{(d)}$ érték az $I_1^{(u)}$ értékét veszi fel $\Delta\tau_1$ késleltetéssel. A blokkvégrehajtóban egy darab monitor csatorna van, ami a késleltetetlen intenzitásokat összegzi blokkonként (M_s). A szimmetrikus normalizálás közelítéséhez ezt az egy értéket használjuk fel.

A blokkvégrehajtó egyetlen korrelátorblokk áramköreit tartalmazza. Ezzel a megoldással az implementált korrelációs csatornák száma L lesz függetlenül az elérni kívánt maximális időfelbontás értékétől. A korrelátorblokk végrehajtási egységet és az ütemezőt együtt korrelációt teljesítő egységnek nevezzük (Correlation Processing Element).

3.4. Correlation Processing Element (CorrPE)

A blokkvégrehajtó és az ütemező együttes használatával számíthatjuk ki a korrelációs függvényt a multiple- τ séma szerint. A két egységet együtt korrelációt teljesítő egységnek (Correlation Processing Element, CorrPE) nevezzük. A CorrPE megvalósításához a blokkvégrehajtó és az ütemező egy ciklusban van beágyazva. A ciklus egy iterációja alatt a blokkvégrehajtó végrehajt egy korrelátorblokkot. Ahhoz, hogy az ütemező ki tudja számolni, hogy a korrelációs függvénynek mely szegmensét kell kiszámolni, egy számlálót használunk, amit c -vel jelölünk a továbbiakban. A következő alfejezetek a CorrPE megvalósításának két típusát mutatják be. A CorrPE-1 terv egy autokorrelációs függvény kiszámítását, a CorrPE-2X terv pedig két auto- és két keresztkorrelációs függvény (2ACF, 2XCF) kiszámítását mutatja be.



3.3. ábra. CorrPE-1 és a CorrPE-2 tervrajzok sematikus ábrája. A blokkvégrehajtó az akkumulátoroktól kapja a bemenő értékeket megfelelő időközönként. Az ütemező határozza meg, melyik korrelátorblokkot kell végrehajtani az adott időegységben. A blokkvégrehajtó a korrelációs adatok tárolására az FPGA memóriaegységeit használja. Az adatok host számítógépre való küldésére a DMA-FIFO-t használjuk.

3.4.1. CorrPE-1

Az egy bemenőjelhez tartozó CorrPE esetében a c értékét minden ciklus után inkrementáljuk. Egy ciklusiterációban a CorrPE végrehajtja az alábbi műveleteket:

- 1. Az ütemező kiszámolja a_c értékét a (3.2)-es egyenletnek megfelelően, majd továbbküldi a végrehajtandó blokk sorszámát ($s = a_c + 1$) a blokkvégrehajtónak.
- 2. A blokkvégrehajtó betölti az s -edik korrelátorblokk paramétereit ($G_{s,l}, M_{s,0}, I_{s,j}^{(d)}, I_s^{(u)}, I_s^{(u)*}, T_s$) a memóriából.
- 3. A blokkvégrehajtó végrehajtja az s -edik korrelátorblokkot.
- 4. c értéke inkrementálódik.

Amennyiben a c -edik ciklusban (időegységben) az ütemező által meghatározott érték nagyobb mint a blokkok száma ($a_c > S$) vagy a kiszámolt korrelátorblokk (s) megelőzője még nem hajtottott végre $(L + m)$ -szer, úgy a 2-es és a 3-as lépések kimaradnak, ami az ún. üres végrehajtást jelent.

Ki kell emelnünk, hogy a (3.2)-es egyenletben definiált sorozat akkor tesz eleget a multiple- τ koncepcióban támasztott feltételeknek (az s -edik korrelátorblokk akkor hajtottott végre, ha az $(s - 1)$ -edik blokk végrehajtódott már kétszer), ha a c számláló 3-tól kezdődik. Az $a_2 = 1$ -től kezdődően a 0 érték kétszer fog előfordulni az 1-es érték legközelebbi előfordulásáig.

Az s -edik blokk akkor szolgáltat adatokat az $(s + 1)$ -edik blokknak ha az s -edik blokk már legalább $(L + m)$ -szer végrehajtódott. A következő képlet megadja, hogy egy adott s blokkot hány ciklusszám elteltével lehet végrehajtani először.

$$c^{(\text{threshold})} = 19 \cdot 2^{s-1} - 16 \quad (3.4)$$

A fenti formulát használjuk arra, hogy megállapítsuk, végre lehet-e hajtani az ütemező által kiválasztott s -edik blokkot az adott ciklusban, vagy pedig egy üres végrehajtás következik. A következő táblázat tartalmazza a CorrPE-1 végrehajtási rendjét.

c	3	4	5	6	7	8	9	10	11	12	13	14	15	16
a_c	0	2	0	1	0	3	0	1	0	2	0	1	0	4
Végrehajtott blokk	1	-	1	-	1	-	1	-	1	-	1	-	1	-
c	17	18	19	20	21	22	23	24	25	26	27	28	29	30
a_c	0	1	0	2	0	1	0	3	0	1	0	1	0	1
Végrehajtott blokk	1	-	1	-	1	2	1	-	1	2	1	-	1	-
c	31	32	33	34	35	36	37	38	39	40	41	42	43	44
a_c	0	5	0	1	0	2	0	1	0	3	0	2	0	1
Végrehajtott blokk	1	-	1	2	1	-	1	2	1	-	1	-	1	2

3.1. táblázat. CorrPE-1 végrehajtása az ütemezésnek megfelelően. A kötőjelek (–) az üres végrehajtásokat jelölik, a végrehajtás a $c = 3$ -tól kezdődik.

3.4.2. CorrPE-2X

Két különböző detektor által mért fluoreszcencia intenzitások fluktuációiból két auto- és két keresztkorrelációt tudunk kiszámolni. A korrelációs számítást a következő képlet segítségével végezzük el (minden x és y párosításra):

$$\hat{g}_{\text{multi-}\tau, \text{symm}}^{(x,y)}(\tau_{s,l}) = \frac{\frac{1}{T_s-l} \cdot \sum_{i=0}^{T_s-1} G_{s,l}^{(x,y)}}{\frac{1}{T_s} \cdot \sum_{i=0}^{T_s-1} M_s^{(x)} \cdot \frac{1}{T_s-l} \cdot \sum_{i=l}^{T_s} M_{s,l}^{(y)}} \quad (3.5)$$

Ahol x és y különbözteti meg az detektorokat (illetve a tőlük származó jeleket). A két változó lehetséges párosításainak száma 4. Abban az esetben amikor a párosítás (x, x) vagy (y, y) , autokorrelációt, egyébként keresztkorrelációt számolunk. Két ACF és két XCF multiple- τ koncepció szerinti kiszámításához $4 \cdot S$ db blokkra van szükség. Az x -szel és y -nal megjelölt APD detektoroktól jövő jeleket a (3.5) egyenletnek megfelelően x és y párokra számoljuk ki a következő sorrendben: (x, x) ; (y, y) ; (x, y) ; (y, x) . Két bemenőjelhez két akkumulátor

van hozzárendelve, amely a különböző APD-ktől érkező intenzitás értékeket akkumulálja és meghatározza a késleltetetlen intenzitás értékeket, $I_s^{(u,x)}$ -t és $I_s^{(u,y)}$ -t. A korrelációs függvények kiszámolt értékeit ($G_{s,l}^{(x,x)}$, $G_{s,l}^{(y,y)}$, $G_{s,l}^{(x,y)}$, $G_{s,l}^{(y,x)}$) az intenzitásokat ($I_{s,l}^{(d,x)}$, $I_{s,l}^{(d,y)}$, $I_s^{(u,x)}$, $I_s^{(u,y)}$) a monitor csatornák értékeit ($M_s^{(x)}$, $M_s^{(y)}$) és a blokk végrehajtási ciklusainak számát (T_s) az FPGA memóriában tároljuk.

A CorrPE-2X megvalósításánál a négy függvény szekvenciális kiszámításához egyetlen blokkvégrehajtót használunk. Ennek eléréséhez az ütemező által meghatározott s -edik blokkot mind a négy korrelációs függvénynél ki kell számolni, ezért $4 \cdot \Delta\tau_{lin}$ időre van szükség mind a négy korrelációs függvény adott s -edik szegmensének kiszámításához. Annak eldöntésére, hogy melyik korrelációs függvényt kell kiszámolni adott ciklusidőben, egy periódikus számlálót használunk (c_{corr}) a $[0, 1, 2, 3]$ tartományban, amely minden $\Delta\tau_{lin}$ időegységben inkrementálódik. Amennyiben c_{corr} értéke 0 vagy 1 akkor autokorrelációs függvényeket számolunk (x, x) ; (y, y) ha c_{corr} értéke 2 vagy 3 akkor a kiszámolt értékek a keresztkorrelációs függvények (x, y) ; (y, x) . A c_{corr} értékének megfelelően meghatározott x és y párokhoz rendelt bemenőjelekre a CorrPE-2X a következő műveleteket hajtja végre egy $\Delta\tau_{lin}$ idő alatt:

- 1. Az ütemező kiszámolja a_c értékét a (3.2)-es egyenletnek megfelelően, majd továbbküldi a végrehajtandó blokk sorszámát ($s = a_c + 1$) a blokkvégrehajtónak.
- 2. A blokkvégrehajtó betölti az s -edik korrelátorblokk paramétereit x és y bemenőjelnek megfelelően ($G_{s,l}^{(x,y)}$, $M_s^{(x)}$, $I_{s,j}^{(d,x)}$, $I_s^{(u,y)}$, $I_s^{(u,y)*}$, T_s) a memóriából.
- 3. A blokkvégrehajtó végrehajtja az s -edik korrelátorblokkot.
- 4. c_{corr} értéke inkrementálódik.

Amennyiben a 4-es pont végrehajtása után c_{corr} értéke 4, úgy a c értéke inkrementálódik, c_{corr} értéke pedig visszaállítódik 0-ra. Az autokorrelációs függvények teljesítése esetén a blokkvégrehajtó a leírásban szereplő összes utasítást végrehajtja, a keresztkorrelációs függvények kiszámítása esetében kihagyja az a) pontban leírt intenzitás csatornák késleltetését. A következő oldalon található táblázat tartalmazza CorrPE-2X szerinti végrehajtási rendjét.

c_{corr}	0	1	2	3	0	1	2	3
Kiválasztott korrelációs függvény	A_x	A_y	X_{xy}	X_{yx}	A_x	A_y	X_{xy}	X_{yx}
c	30	30	30	30	31	31	31	31
a_c	1	1	1	1	0	0	0	0
Végrehajtott szegmens	2	2	2	2	1	1	1	1

c_{corr}	0	1	2	3	0	1	2	3
Kiválasztott korrelációs függvény	A_x	A_y	X_{xy}	X_{yx}	A_x	A_y	X_{xy}	X_{yx}
c	32	32	32	32	33	33	33	33
a_c	5	5	5	5	0	0	0	0
Végrehajtott szegmens	-	-	-	-	1	1	1	1

3.2. táblázat. CorrPE-2X végrehajtása az ütemezésnek megfelelően. A kötőjelek (-) az üres végrehajtásokat jelölik. A c értékét a táblázatban 30-al kezdtük. Az A_x , A_y , X_{xy} , X_{yx} jelölik a végrehajtásra kiválasztott korrelációs függvényt c_{corr} értékének megfelelően az adott ciklusiterációban. A kötőjelek (-) az üres végrehajtást jelölik.

3.5. A kiszámított korrelációs értékek kiolvasása és normalizálása

A CorrPE-1 esetben a kiszámolt $(G_{s,l}, M_s, I_{s,j}^{(d)}, I_s^{(u)}, I_s^{(u)*}, T_s)$ értékekből a korrelációs függvény meghatározható. Az ehhez szükséges lebegőpontos műveleteket a host számítógépen végezzük el. Az általunk megvalósított hardware korrelátor az s -edik blokk ($s > 1$) számításait úgy ütemezte, hogy azt az $(s - 1)$ -edik korrelátorblokk tétlen idejeiben számolja ki.

A korrelátorblokkok paramétereinek a kiolvasást úgy oldottuk meg, hogy a c számláló $(2^S - 1) \cdot k \cdot \tau_{\min}$, $k \in \mathbb{N} \setminus \{0\}$ értékeinél a soron következő s -edik korrelátorblokk $G_{s,i}$, M_s , és T_s értékeit kiolvastuk egy arra dedikált memóriarészbe, majd ezeket az értékeket 0-ra állítottuk. A kiolvasásra használt memóriában tárolt értékeket a hardware korrelátor az s -edik blokk következő kiolvasása előtt a DMA-FIFO-ba „küldi”, ahonnan egy erre a célra kifejlesztett alkalmazás azt a host számítógépen tárolja.

A szimmetrikus normalizáláshoz ((2.14)-es egyenlet) minden $\tau_{s,l}$ momentum $G_{s,l}$ értékéhez két monitor csatorna értéket kell használnunk: M_s és $M_{s,\tau_{s,l}}$. Azonban ha $T_s \rightarrow \infty$ a $\lim_{T_s \rightarrow \infty} M_{s,\tau_{s,l}} = \lim_{T_s \rightarrow \infty} M_s$. Az $M_{s,\tau_{s,l}}$ becslését a következő képlettel adhatjuk meg:

$$M_{s,\tau_{s,l}} \sim \frac{T_s - (l - 1)}{T_s} \cdot M_s \quad (3.6)$$

Ennek előnye, hogy blokkonként egy monitorcsatornát kell implementálnunk. A CorrPE-2X

esetében a normalizálás a (3.5) egyenletnek megfelelően hajtódik végre az x és y detektorokból jövő akkumulált intenzitásokra.

Az értékek periodikus kiolvasása és a normalizálás után lehetővé válik a korrelációs görbe megjelenítése gyakorlatilag a méréssel egyidejűleg. A következő fejezetben a megvalósításhoz szükséges technikai részleteket mutatom be.

4. Multiple- τ korrelátor implementálása

4.1. FPGA használata digitális korrelátorok megvalósításához

Az FPGA (field programmable gate array) egy programozható logikai kapumátrixokat tartalmazó hardware. Legfőbb jellegzetessége a programozhatósága, amellyel egy általános célú digitális eszköz megvalósítható rajta. Szemben az ASIC (Application Specific Integrated Circuits) eszközökkel, melyek logikai áramköreit a gyártás után már nem lehet változtatni, az FPGA eszközök kapumátrixainak logikai áramköreit újratervezhetjük az adott célnak megfelelően. További különlegessége, hogy kiválóan alkalmas párhuzamos számítások elvégzéséhez. A legtöbb FPGA tartalmazza a következő elemeket

- I/O blokkok: interfészek, amelyeken keresztül az eszköz kapcsolatot tart a külvilággal.
- Core: Belső programozható rész
- BlokkRAM: SRAM az adatok tárolására.

A kapumátrixokban az általunk használni kívánt logikát leírónyelvekkel definiálhatjuk. A két legismertebb nyelv FPGA-khoz a VHDL (very-high-speed integrated circuits hardware description language), illetve a Verilog.

Az elmúlt években több folyóiratban publikált közlemény szólt az FPGA használhatóságáról korrelációs számításokhoz ([2, 3, 4]), azonban a megvalósításhoz szükséges részletek tisztázása nem kellőképpen részletes, vagy a megvalósíthatóságra csak utalások vannak a különböző cikkekben. Az FPGA megvalósítás során az implementálandó korrelációs, monitor és intenzitási csatornák száma korlátozott. A kapumátrixok által definiált logika, a felhasználásra szánt flip-flopok mind-mind korlátozottak. A korábbi fejezetben utalt 194 korrelációs csatorna implementálása a multiple- τ koncepció megvalósításához FCS mérések esetén sok esetben nem kivitelezhető FPGA-ra a felhasznált flip-flop-ok nagy száma miatt. A kidolgozott eljárás lecsökkentette az implementálandó korrelációs csatornák számát 8-ra amely már a legtöbb

forgalomban lévő FPGA-val megvalósítható.

A hardware korrelátor programjának elkészítéséhez a National Instruments (NI) cég által fejlesztett LabView szoftver 8.6-os verzióját az FPGA module kiegészítővel használtuk. Az elkészített hardware korrelátorok (CorrPE-1, CorrPE-2X) a felhasznált erőforrások tekintetében csekély mértékben különböznek.

4.2. CorrPE-1 és CorrPE-2X hardware korrelátorok

A használt FPGA egy Virtex-II (XC2V3000, Xilinx, <http://www.xilinx.com/>) chip. Az FPGA egy NI (National Instruments) PXI-7833R kártyán érhető el, a programozáshoz LabView 8.6-os verzióját használtuk az FPGA modul kiegészítővel.

A használt korrelációs csatornák száma (L) 8. A korrelátor 20 MHz-es mintavételezési frekvenciával rendelkezik, így az $\tau_{\min}$ értéke $50ns$. A CorrPE-1 esetében a minimális időfelbontás $100ns$ és a korrelátorblokkok száma (S) 20, így $\tau_{\max} = 838,86ms$. A CorrPE-2X esetében a minimális időfelbontás $400ns$, az S értéke pedig 18.

CorrPE-1 esetében a korrelátorblokkok paramétereinek ($G_{s,l}$, M_s , $I_{s,j}^{(d)}$, $I_s^{(u)}$, $I_s^{(u)*}$, T_s) tárolásához szükséges egymástól független memóriarekeszek száma 31. A "delayed" intenzitások ($I_{s,j}^{(d)}$) tárolásához 10, az "undelayed" intenzitások ($I_s^{(u)}$ és $I_s^{(u)*}$) tárolásához 2, a korrelációs csatornák által kiszámított értékekhez ($G_{s,l}$) 8, a monitorcsatorna értékének (M_s) tárolásához 1, a blokkonkénti mintavételezés számának (T_s) tárolásához 1, a kiolvasáshoz pedig 9 memóriarekesz szükséges. A kiolvasott és aktuális $G_{s,l}$ értékeket 64, a többi értéket 32 biten reprezentáljuk a memóriában. A megcímezhető értékek száma memóriarekeszenként 32. A használt Virtex-II kártyán a 64 bites reprezentáció két 32 bites memóriablokkot foglal el, így a korrelációs számításra felhasznált memóriablokkok száma 48. A következő táblázat összefoglalja a felhasznált memóriablokkok számát.

Memóriablokk felhasználása	Reprezentáció	Szükséges memóriarekeszek száma	Tárolásra felhasznált RAMB16_S36_S36-ok száma
Kiolvasás	64-bit	9	$2 \cdot 9$
$G_{s,l}$	64-bit	8	$2 \cdot 8$
$I_{s,j}^{(d)}$	32-bit	10	10
$I_s^{(u)}$	32-bit	1	1
$I_s^{(u)*}$	32-bit	1	1
M_s	32-bit	1	1
T_s	32-bit	1	1
Összesen	0	31	48

4.1. táblázat. Táblázat A RAMB16_S36_S36 egy Virtex-II által használt 32 bites dual-port elérésű RAM. A 64 bites reprezentációhoz 2db 32 bites RAMB16_S36_S36-ra van szükség.

A kiolvasáshoz használt 9 memóriarekeszből 8 direkt módon egy-egy $G_{s,l}$ értékhez kapcsolódik, míg az M_s , T_s 32 bites értékek összefésülődnek egyetlen 64 bites értéké, amely a 9. kiolvasásra használt memóriarekeszben tárolódik. A teljes CorrPE-1 a DMA-Engine-el együtt 60 memóriablokkot használ fel az elérhető 96-ból (62%). A korrelációs számításhoz felhasznált memóriaterület nagysága $48 \cdot 32 \cdot 32 \text{ bit} = 6 \text{ Kbyte}$.

A CorrPE-2X megvalósításnál a memóriarekeszek száma megegyezik a fentebb (CorrPE-1-hez) leírtakkal. Különbség a felhasznált memóriaterület méretében van. Két bemenő jel intenzitásainak tárolásához az intenzitás csatornákhöz és a monitorcsatornákhöz rendelt memóriaterület megduplázódik, a korrelációs számítási értékek tárolására használt memóriaterület pedig megnégyszereződik. A blokkonkénti mintavételezések számának tárolása nem változik, mivel a korrelációs függvények azonos szegmensében a korrelátorblokkok mintavételezési száma mindig meg kell, hogy egyezzen. A konkrét memóriaterület felhasználása a következőképpen alakul:

$(I_s^{(u,x)}, I_s^{(u,y)}; I_s^{(u,x)*}, I_s^{(u,y)*}; I_{s,i}^{(d,x)}, I_{s,i}^{(d,y)}; M_s^{(x)}, M_s^{(y)} : 13 \cdot 2 \cdot 32 \cdot 32 \text{ bit} = 3,25 \text{ Kbyte}; G_{s,l}^{(x,x)}, G_{s,l}^{(y,y)}, G_{s,l}^{(x,y)}, G_{s,l}^{(y,x)} : 16 \cdot 4 \cdot 32 \cdot 32 \text{ bit} = 8 \text{ Kbyte};$ a kiolvasáshoz szükséges RAM-ok száma: $18 \cdot 4 \cdot 32 \cdot 32 \text{ bit} = 9 \text{ Kbyte}$, illetve a T_s , ami $1 \cdot 32 \cdot 32 \text{ bit} = 0,125 \text{ Kbyte}$; ez összesen $20,375 \text{ Kbyte}$.

A következő táblázat tartalmazza a CorrPE-2X lefordítása után a felhasznált erőforrásokat.

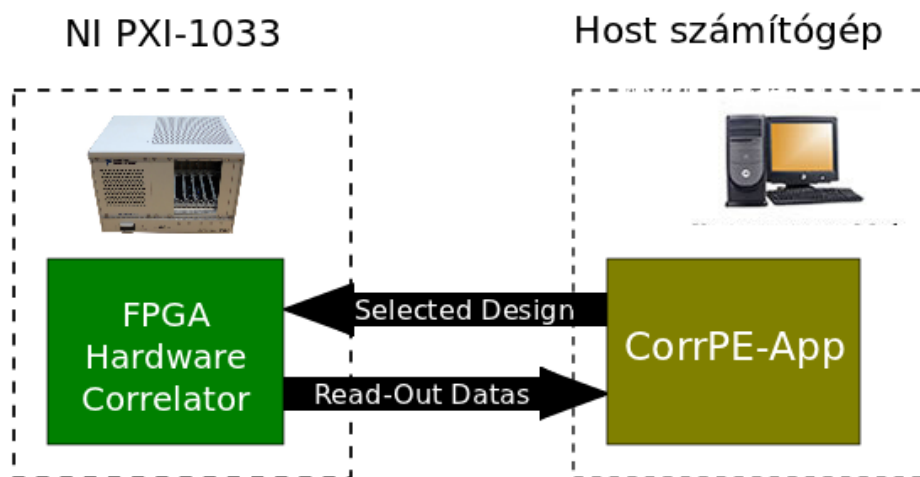
Elem neve	Felhasznált	Összes	Kihasználtság
Number of Slices	5318	14336	37%
Number of BRAMs	60	96	62%
Number of MULT18X18s	45	96	46%

4.2. táblázat. A CorrPE-2X teljes kihasználtsága.

A PXI-7833R kártya egy PXI-1033-as rendszerbe van beágyazva, a host computerhez egy PCI Express kártyán keresztül csatlakozik. A host számítógépen a design FPGA kártyára való feltöltéséhez, a hardware korrelátortól jövő adatok kiolvasásához és a korrelációs görbe megjelenítéséhez egy erre a célra kifejlesztett alkalmazást készítettünk el, amelyet CorrPE-App-nak nevezünk.

4.3. CorrPE-App

A CorrPE-App egy Microsoft Visual Studio 2008 környezetben kifejlesztett, C++ nyelven megírt alkalmazás. A program az FPGA használatához az NI által szolgáltatott dinamikus csatolású előre lefordított programkódot használ, az NiFPGA.dll állományt. A megfelelő eljárások meghívásával, a program feltölti a kiválasztott hardware korrelátort (CorrPE-1, CorrPE-2X) az FPGA kártyára, majd a kiolvasott adatokat feldolgozza és megjeleníti. A grafikus felhasználói felületet MFC (Microsoft Foundation Class Library) segítségével valósítottuk meg, a görbék kirajzolásához NTGraph[10] nyílt forráskódú ActiveX komponenst használtunk. A következő ábra a rendszer sematikus felépítését szemlélteti.



4.1. ábra. Az adatfolyam útja a hardware korrelátortól a normalizált korrelációs függvény megjelenítéséig. A hardware korrelátortól a DMA-FIFO-n keresztül juttatjuk el az adatokat a CorrPE-App alkalmazáshoz, amely megjeleníti a kiszámolt korrelációs függvényt, elmenti a nyers adatokat és exportálja a kiszámított értékeket.

5. Eredmények

Az FCS/FCCS mérésekhez egy Olympus FV1000 konfokális mikroszkópot használtunk egyedileg hozzátervezett FCS/FCCS egységgel (3db EGG APD detektor).

A CorrPE-1 és CorrPE-2X korrelátorok teszteléséhez Alexa488 és Cy5 festékek oldatait használtuk. A festékeket 488, illetve 633 nm-es lézerfényvel gerjesztettük. Az emittált fotonokat egy 570DXCR dikroikus tükör választotta szét hullámhossz szerint, majd a zöld tartományba eső fotonokat egy 514/20 nm-es, a vörös tartományba eső fotonokat egy 690/70 nm-es sávszűrőn keresztül detektáltuk az APD 0 és 1-el. A mérések időtartama $5 \cdot 20$ másodperc volt.

A korrelátor teszteléséhez a detektorokból jövő TTL jeleket megosztottuk az FPGA hardware korrelátor és egy ALV5000E (ALV GmbH, Langen, Germany) multiple- τ hardware korrelátor között. A CorrPE-2X által meghatározott autokorrelációs görbe teszteléséhez 50nM-os Alexa 488 (zöld fluoreszcenciájú) és Cy5 (vörösben és IR-ben emittáló) festékoldatokat használtunk. A két festékoldat CorrPE-2X el és ALV-5000E-vel számolt autokorrelációs függvényeket az (

Egy ACF göbe kiszámítása esetében a megillesztésre (2.2)-es és (2.5)-es egyenletben leírt modellfüggvényt használtuk. Az autokorrelációs görbék modellfüggvénye a duplán jelölt antitestek esetében két diffúziós komponenst tartalmazott (gyors, antitesthez nem kötődő és lassú, antitesthez kötődő komponens), és figyelembe vette a triplet állapotot:

$$g(\tau) = \frac{1}{N} \frac{1 - T + T e^{\tau/\tau_r}}{1 - T} \cdot \left[r_1 \left(1 + \frac{\tau}{\tau_1} \right) \left(1 + \frac{\tau}{\tau_1 \cdot S^2} \right) + r_2 \left(1 + \frac{\tau}{\tau_2} \right) \left(1 + \frac{\tau}{\tau_2 \cdot S^2} \right) \right] \quad (5.1)$$

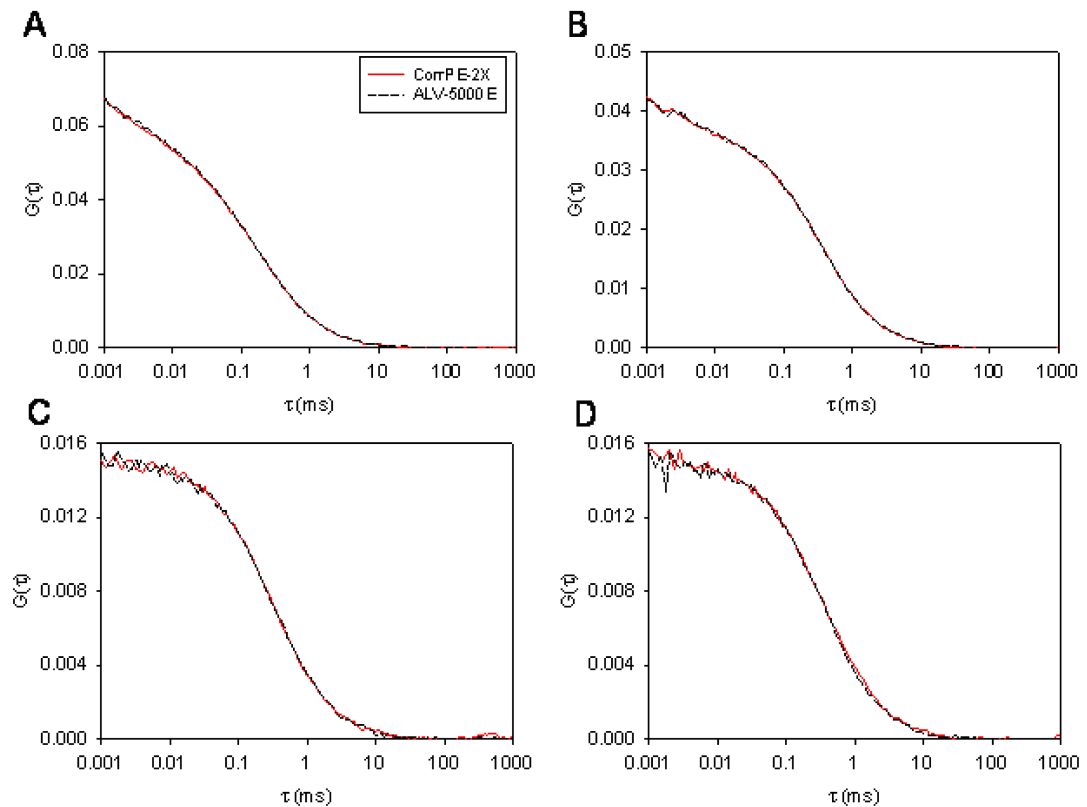
Ahol r_1 és $r_2 = 1 - r_1$ a gyors és lassú komponensek fraktális amplitúdója, a komponensek diffúziós ideje pedig τ_1 és τ_2 . A keresztkorrelációs görbék megillesztésénél a triplet állapotot figyelmen kívül hagytuk, mivel a kétféle festék triplet állapotba kerülése egymástól függetlenül következik be, így az általuk okozott ingadozások keresztkorrelációja zérus.

Az ACF görbéket leíró modellfüggvény az (2.1) egy modell függvénnyel voltak megillesztve amely figyelembevétel a triplet állapotot és egy diffúziós komponenst a következőképpen van

definiálva:

$$g(\tau) = \frac{1}{N} \frac{1 - T + T e^{\tau/\tau_r}}{1 - T} \cdot r_1 \left(1 + \frac{\tau}{\tau_1} \right) \left(1 + \frac{\tau}{\tau_1 \cdot S^2} \right) \quad (5.2)$$

A modellfüggvények megillesztéséhez a Jan Krieger (Ph. D. hallgató, Biophysics of Macromolecules, DKFZ, Heidelberg) által fejlesztett Quickfit 3.0 programot használtuk.



5.1. ábra. Az ábrán alexa Fluor 488 és Cy5 festékekkel duplán jelölt W6/32-es antitestek auto- és keresztkorrelációs görbéi láthatóak. A számítást a CorrPE-2X (piros egyenes vonal) és egy ALV5000 (fekete szaggatott vonal) korrelátorokkal végeztük. Az A és B ábra a zöld és a piros bemenetekhez kapcsolt autokorrelációs görbét ábrázolja, míg a C és a D ábra a két bemenet között számolt keresztkorrelációt jeleníti meg.

6. Összefoglalás

Az elkészített hardware korrelátor szimultán képes FCS és FCCS mérésekhez korrelációs függvények kiszámítására, amely használható az intézet sejtbiológiai kutatásaihoz. A megjelölt molekulák auto- és keresztkorrelációs függvényeinek meghatározása komplex biofizikai jellemzést tesz lehetővé, amely információt ad a molekulák mobilitásával, a diffúzió mechanizmusával, illetve a kétféle molekula egymással és a környezet egyéb molekuláival kialakított kölcsönhatásaival kapcsolatban. A Biofizikai és Sejtbiológiai Intézet munkatársai a korrelátort a sejtmembrán és a sejtmag fehérjéinek vizsgálatára fogják felhasználni.

A szimultán FCS és FCCS mérést a jelenleg kapható hardware korrelátorok többsége nem támogatja (Forrás: <http://www.correlator.com>, <http://alvgmbh.com>). A megvalósítása során kifejlesztett algoritmus minimalizálja az implementált korrelációs csatornák számát. A multiple- τ koncepció alapján az implementálandó korrelációs csatornák száma $S \cdot L$ -ről L -re csökkenthető, ami alkalmassá teszi a korrelátor implementálását a legtöbb FPGA kártyán. További előnye, hogy az implementált korrelációs csatornák száma konstans marad függetlenül az elérni kívánt maximális időfelbontás növekvő értékétől. A maximális időfelbontás az implementálandó áramkörök kialakítását nem befolyásolja, a felbontás növelése az adattár növekedését vonja maga után. A multiple- τ koncepció ezen megvalósítása mindezig nem publikált eljárás. Az ütemezési eljárást tovább lehet fejleszteni más periodikus folyamatok végrehajtásához kapcsolódó teljesítménybeli problémák optimalizálására. (pl. napi, heti és havi biztonsági mentések ütemezése, állandóan és periodikusan futó folyamatok ütemezése, stb.).

A megvalósított multiple- τ hardware korrelátor egy (CorrPE-1), illetve négy (CorrPE-2X) korrelációs függvény szimultán kiszámítására alkalmas. A minimális időfelbontás a CorrPE-1 esetében 100ns, a CorrPE-2X esetében 400ns. A maximális időkésleltetés mindkét esetben 838,86ms. Az FPGA korrelátor vezérlése a C programnyelven megírt FPGA Reader program végzi amely felelős a kiolvasott adatok tárolásáért is. Az AXCF Controller .NET keretrendszerben íródott, amelynek eszközrendszerével könnyedén kialakíthatunk rugalmas felhasználói felületeket. Az adatok tárolására egy PostgreSQL adatbázist használunk, amelyben az adatok szűrése és statisztikai kiértékelése válik könnyen kezelhetővé.

6.1. Saját munka leírása

A Debreceni Egyetem Orvos- és Egészségtudományi Centrum Biofizikai és Sejtbiológiai Intézetének Sejtbiológiai és Jelátviteli Kutatócsoportjában 2010 februárjában kezdtem dolgozni Dr. Vámosi György és Mocsár Gábor témavezetésével. Munkám során felelős voltam a hardware korrelátor elkészítéséért és a megvalósítás során felmerülő problémák megoldásáért. Felismertem, hogy a korrelátor végrehajtása a multiple- τ koncepció alapján optimalizálható, megadtam egy sorozatot amely alapján a korrelátorblokkok ütemezését megoldottuk. Ennek az eljárásnak köszönhetően csökkentettük az implementálandó korrelációs csatornák számát, mellyel lehetővé vált az implementálás a használt FPGA kártyára. Megadtam a sorozattal ekvivalens formulát mely használatával a sorozat értékeinek kiszámításához nem szükséges rekurzió vagy nagyméretű lookup tábla. Bebizonyítottam, hogy a sorozat alkalmas a multiple- τ koncepcióban támasztott feltételek kielégítésére. Megterveztem és implementáltam a CorrPE-1 és CorrPE-2X hardware korrelátorokat és megoldottam, hogy a négy korrelációs függvényt egy blokkvégrehajtó implementálásával valósítsuk meg a CorrPE-2X korrelátorban. A kiértékeléshez használt program tervezése, megvalósítása és tesztelése szintén munkám részét képezték.

6.2. Köszönetnyilvánítás

Tisztelettel szeretnék köszönetet mondani témavezetőimnek Dr. Vámosi Györgynek és Mocsár Gábornak, akik lehetővé tették számomra a munka elkészítéséhez. Köszönettel tartozom továbbá Dr. Tóth Lászlónak (egyetemi adjunktus, Debreceni Egyetem Informatika kar Informatikai Rendszerek és Hálózatok Tanszék), Jan Kriegernek (Ph.D hallgató, dkfz Heidelberg B040), Jan Buchholznak (Ph.D hallgató, dkfz Heidelberg B040) valamint Panyi Györgynek a Debreceni Egyetem Orvos- és Egészségtudományi Centrum Általános Orvostudományi Kar Biofizikai Tanszék vezetőjének.

A. Függelékek

A.1. A (3.2)-es egyenlethez kapcsolódó állítások és bizonyítások

Definiáljuk az a_c sorozatot a következőképpen:

$$a_c = \begin{cases} 0 & \text{ha } c \text{ páratlan vagy } 0 \\ a_{\frac{c}{2}} + 1 & \text{egyébként} \end{cases}, c = 0, 1, \dots \quad (\text{A.1})$$

Annak belátásához, hogy a (3.2)-es egyenletben definiált sorozat sorban növekvő c ($c \in \mathbb{N}$) számokra pontosan a végrehajtandó korrelátorblokk sorszámát adja meg, igazolnunk kell, hogy a sorozat periodikus minden $a_c = k$, $k \in \mathbb{N}$ értékre, valamint bármely ($k > 0$) érték megjelenése után a $k - 1$ érték pontosan kétszer jelenik meg a sorozatban a k érték következő megjelenéséig, azaz a $k - 1$ korrelátorblokk kétszer lesz végrehajtva k -adik blokk legközelebbi végrehajtásáig. Az állítás bizonyításhoz szükségünk lesz az alábbi definíciókra és lemmákra:

Definíció: Legyen $a_c = k$, ahol $c, k \in \mathbb{N}$, és legyen $d > c$ a c -hez legközelebbi olyan szám, amelyre teljesül $a_d = k$.

A.1.1. Lemma: Legyen $a_c = k$. Akkor $c = n \cdot 2^k$, ahol n egy tetszőleges páratlan természetes szám.

Bizonyítás: Minden $c \in \mathbb{N}$ felírható $c = n \cdot 2^k$ alakban, és a_c értéke az (A.1)-es egyenletben leírt sorozat definíciójából következően k lesz.

A.1.2. Lemma: Legyen $a_c = k$. A k érték előfordulása az (A.1)-es egyenletben definiált sorozatban periódikus és a periodicitása 2^{k+1} , azaz $a_c = a_{c+2^{k+1}} = k$.

Bizonyítás: Legyen $c = n \cdot 2^k$, ahol n egy tetszőleges páratlan természetes szám. A következő páratlan szám $n + 2$, tehát a következő index, ahol a sorozat értéke k lesz az $(n + 2) \cdot 2^k$ helyen lesz, ami $c + 2^{k+1}$.

A.1.3. Lemma: Legyen $a_c = k$, ahol $k > 0$, n egy páratlan természetes szám, és $c = n \cdot 2^k$. Legyen $l = c + 2^{k-1}$. Ekkor $a_l = k - 1$.

Bizonyítás: Az $l = n \cdot 2^k + 2^{k-1}$, felírható $l = (2 \cdot n + 1) \cdot 2^{k-1}$ alakban, ahol $2 \cdot n + 1$ biztos, hogy páratlan, tehát az a_l értéke az A.1.1.-es lemmából következően $k - 1$ lesz.

A.1.4. Lemma: Legyen $a_c = k$, ahol $k > 0$, n egy páratlan természetes szám, $c = n \cdot 2^k$, és legyen $l = c + 2^{k-1}$, az A.1.3. lemmából következően $a_l = k - 1$. Az l a legközelebbi következő index c -hez, ahol az (A.1)-es egyenletben definiált sorozat értéke $k - 1$ lesz,

Bizonyítás: Indirekt. Tegyük fel, hogy van olyan $g \in \mathbb{N}$, hogy $c < g < l$ és $a_g = k - 1$. Akkor az A.1.1.-es lemmából következően g felírható $h \cdot 2^k - 1$ alakban, ahol h páratlan természetes szám. Ekkor $A \ c < g$ és $g < l$ egyenlőtlenségeknek igaznak kell lenniük. A $c < g$ egyenlőtlenségből azt kapjuk, hogy $2 \cdot n < h$, a $g < l$ egyenlőtlenségből pedig azt, hogy $h > 2 \cdot n + 1$ ami ellentmondás.

Állítás: Létezik pontosan két $m, p \in \mathbb{N}$ index c és d között, hogy $a_m = a_p = k - 1$, azaz $c < m < p < d$.

Bizonyítás: Az A.1.2.-es lemmából következően $d = c + 2^{k+1}$. Legyen m értéke a c -hez legközelebbi olyan szám, amelyre teljesül, hogy $a_m = k - 1$. Az A.1.4.-es lemmából következően $m = c + 2^{k-1}$, az A.1.2.-es lemmából következően $p = m + 2^k$. Ekkor, az, hogy $c < m$ valamint az $m < p$ triviális. A $p < d$ egyenlőtlenségből azt kapjuk, hogy $2 \cdot n + 3 < 2 \cdot n + 4$ ami igaz. Legyen w egy index úgy, hogy $p < w < d$ és $a_w = k - 1$. A p -hez legközelebbi olyan index, ahol a sorozat a $k - 1$ értéket veszi fel a $p + 2^k$ helyen lesz. A $w < d$ egyenlőtlenségből azt kapjuk, hogy $2 \cdot n + 5 < 2 \cdot n + 4$ ami nem igaz. Azaz pontosan két index létezik c és d között, amelyre az állítás teljesül.

A.2. A (3.3)-as egyenlethez kapcsolódó állítások és bizonyítások

Az (A.1)-es egyenletben definiált sorozat megadja, hogy egy adott c természetes szám hányszor osztható 2-vel, ezért a szám bináris formájában meg kell keresnünk az első olyan bitet jobbról balra, amely 1-es. Ennek a számnak a helyiértéke fogja megadni a sorozat aktuális értékét.

$$a_c = \log_2(X_c) - 1, \quad X_c = (c \text{ XOR } c - 1) + 1 \quad (\text{A.2})$$

ahol a XOR egy bitenkénti logikai kizáró vagy művelet a két kijelölt szám között. Az X_c bináris

formájában egyetlen darab 1-es szerepel, így ennek a bitnek a helye jelenti a szám 2-es alapú logaritmusát. Annak belátására, hogy az X_c a megfelelő helyiértéken fog 1-est előállítani bináris formában a következő levezetést adjuk:

Tegyük fel, hogy az első 1-es a c szám bináris formájában a k -adik helyen van (32 bites unsigned integer). Az előlött lévő helyiértékek értéke tetszőleges lehet $c - 1 = (\alpha_3 2^3 \alpha_4 2^4 \dots 0_k 2^k 1 1 \dots 1)_2$.

Ekkor $c \oplus (c - 1) = (00 \dots 0_{k+1} 1_k 1 1 \dots 1)_2$, így $(c \oplus (c - 1) + 1) = (00 \dots 1_{k+1} 0_k 00 \dots 0)_2$.

A kiszámolt szám bináris formájában az 1-es helyiértéke az eredeti szám bináris formájában szereplő első 1-es bit helyiértékénél pontosan 1-el nagyobb, tehát: $\log_2(X_c) - 1 = (k)_{10}$. Mivel X_c mindig 2-nek a hatványa és ez pontosan 1 db egyest tartalmaz, ennek megkeresésére a következő algoritmust használjuk (Nagy Ferenc debreceni fizikus Ph.D. hallgató ötlete alapján, személyes kommunikáció, 2011.02.17):

Legyen x az a szám, amelynek bináris formájában pontosan 1db 1-es bit helyezkedik el.

1. lépés:

$$m_4 = x \text{ AND } (11111111111111110000000000000000)_2 \quad (\text{A.3})$$

$$m_3 = x \text{ AND } (11111111000000001111111100000000)_2 \quad (\text{A.4})$$

$$m_2 = x \text{ AND } (11110000111100001111000011110000)_2 \quad (\text{A.5})$$

$$m_1 = x \text{ AND } (11001100110011001100110011001100)_2 \quad (\text{A.6})$$

$$m_0 = x \text{ AND } (10101010101010101010101010101010)_2 \quad (\text{A.7})$$

Ahol az AND egy bitenkénti logikai ÉS művelet a két kijelölt szám között.

2. lépés:

$$R_i = \begin{cases} 2^i & \text{ha } m_i > 0 \\ 0 & \text{egyébként} \end{cases} \quad (\text{A.8})$$

A helyiérték értéke: $\sum_{i=0}^4 R_i$.

Irodalomjegyzék

- [1] M. Eigen and R. Rigler. Sorting single molecules: application to diagnostics and evolutionary biotechnology. *Proc Natl Acad Sci U S A*, 91(13):5740–5747, Jun 1994.
- [2] C. Jakob, A. T. Schwarzbacher, B. Hoppe, and R. Peters. A fpga optimised digital real-time multichannel correlator architecture. In *Proc. 10th Euromicro Conf. Digital System Design Architectures, Methods and Tools DSD 2007*, pages 35–42, 2007.
- [3] C. Jakob, A. T. Schwarzbacher, B. Hoppe, and R. Peters. A multichannel digital real-time correlator as single fpga implementation. In *Proc. 15th Int Digital Signal Processing Conf*, pages 276–279, 2007.
- [4] C. Jakob, A. Th. Schwarzbacher, B. Hoppe, and R. Peters. The development of a digital multichannel correlator system for light scattering experiments. In *Proc. IET Irish Signals and Systems Conf*, pages 99–103, 2006.
- [5] J. Langowski. Protein-protein interactions determined by fluorescence correlation spectroscopy. *Methods Cell Biol*, 85:471–484, 2008.
- [6] D. Magde, E. L. Elson, and W. W. Webb. Fluorescence correlation spectroscopy. ii. an experimental realization. *Biopolymers*, 13(1):29–61, Jan 1974.
- [7] P. Schwille, F. J. Meyer-Almes, and R. Rigler. Dual-color fluorescence cross-correlation spectroscopy for multicomponent diffusional analysis in solution. *Biophys J*, 72(4):1878–1886, Apr 1997.
- [8] K. Schätzel. New concepts in correlator design. In *Proc. of the int. Phys. Conference*, volume 77, 1987.
- [9] K. Schätzel. Accuracy of photon correlation measurements on nonergodic samples. *Appl Opt*, 32(21):3880–3885, Jul 1993.
- [10] Nikolai Teofilov. 2d graph activex control. <http://www.codeproject.com/Articles/3214/2D-Graph-ActiveX-Control>, 2003 08. Open-source activex component.

- [11] György Vámosi, Sándor Damjanovich, János Szöllosi, and György Vereb. Measurement of molecular mobility with fluorescence correlation spectroscopy. *Curr Protoc Cytom*, Chapter 2:Unit2.15, Oct 2009.
- [12] Thomas Weidemann, Malte Wachsmuth, Tobias A. Knoch, Gabriele Müller, Waldemar Waldeck, and Jörg Langowski. Counting nucleosomes in living cells with a combination of fluorescence correlation spectroscopy and confocal imaging. *J Mol Biol*, 334(2):229–240, Nov 2003.
- [13] J. Widengren and R. Rigler. Fluorescence correlation spectroscopy as a tool to investigate chemical reactions in solutions and on cell surfaces. *Cell Mol Biol (Noisy-le-grand)*, 44(5):857–879, Jul 1998.