

**Budapesti Műszaki és Gazdaságtudományi Egyetem**

Villamosmérnöki és Informatikai Kar

Elektronikus Eszközök Tanszéke

Mózer Viktor

**DIGITÁLIS HULLÁMFORMA GENERÁTOR MEGVALÓSíTáSA ÉS MÉRÉSE FPGA KÖRNYEZETBEN**

Konzulens

Horváth Péter (BME-EET)

Antus Tibor (Indie Semiconductor Hungary)

BUDAPEST,2021

**Feladatkiírás**

**DIGITÁLIS HULLÁMFORMA GENERÁTOR MEGVALÓSITASA ÉS MÉRÉSE FPGA KÖRNYEZETBEN (INDIE SEMICONDUCTOR HUNGARY)**

A hallgató feladata egy digitális hullámforma generátor megtervezése, megvalósítása és mérése FPGA környezetben.

A feladat első részében a Direct Digital Synthesis elvét kell megismerni a hallgatónak, majd adott specifikáció alapján a hullámforma generátor szimulációs környezetét kell elkészíteni (Matlab, Python, etc.).

A második lépésben a hullámforma generátor RTL kódját, az RTL kód szimulációs környezetét és fizikai megvalósítását (synthesis, PnR, Timing analysis) kell elkészítenie a hallgatónak Altera DE2 board-ra.

Harmadik lépésben az implementációt és a Matlab / Python szimuláció eredményét kell összehasonlítani, az FPGA board-on a DA átalakítás után fizikai jelet visszamérni és a következtetéseket levonni.

[Bevezetés 4](#_Toc89595409)

[Irodalmi áttekintés 5](#_Toc89595410)

[1.1 Függvénygenerátorok 5](#_Toc89595411)

[1.2 Jelek spektruma 6](#_Toc89595412)

[1.3 Mintavételezés 7](#_Toc89595413)

[1.4 Direct Digital Synthesis 10](#_Toc89595414)

[1.4.1 Numerically-Controlled Oscillator (NCO) 11](#_Toc89595415)

[1.4.2 DA átalakítás 16](#_Toc89595416)

[1.5 Digitális tervezés alapjai 18](#_Toc89595417)

[2 célkitűzések 20](#_Toc89595418)

[3 Rendszerterv 21](#_Toc89595419)

[3.1 NCO modul 22](#_Toc89595420)

[3.2 Kommunikáció 24](#_Toc89595421)

[3.3 FSM 25](#_Toc89595422)

[3.4 Dump 25](#_Toc89595423)

[3.5 DAC 26](#_Toc89595424)

[3.6 Reset szinkronizáció 26](#_Toc89595425)

[4 Eredmények értékelése, jövőbeli fejlesztések 27](#_Toc89595426)

[Irodalomjegyzék 28](#_Toc89595427)

Tartalomjegyzék

# Bevezetés

A cél egy a Direct Digital Synthesis elvén működő hullámforma generátor tervezése, szimulálása, létrehozása FPGA kártyán majd ennek tesztelése volt.

Az első fejezetben a Direct Digital Synthesis elvének leírása szerepel. Az egyes részegységek működésének és azok elméleti hátterének magyarázata, ábrákkal és levezetésekkel. Elsőként a mintavételezésre tértem ki, mivel ez szükséges egy adott jel generálásához, amit az NCO hoz létre a memóriában tárolt elemekből. Ezek után magával az NCO elvét írtam le, ami gyakorlatilag a hullámformát létrehozza. A mintavételi tétel alapján mekkora frekvencia szükséges a rendszer működéséhez és ez hogyan függ össze a kimeneti frekvenciával. Továbbá mennyi adatot kell tárolni egy tetszőleges frekvenciatartományban lévő kimeneti jel létrehozásához. Ezt egy python nyelven írt kóddal is szimulálni lehet. Az áramkör kimenete egy analóg jelet ad ki, ezért a digitális-analóg átalakítók működési elvének megértése is fontos a megvalósítás szempontjából. Az is lényeges, hogy ez hogyan illeszthető az NCO-hoz. Mivel a számítógépnek adatokat kell küldenie az áramkörünknek, így kifejtésre került a kommunikációs egység miként választható le a modulról.

A második részben elkészítettem a rendszertervet. Az RTL szintű modell megtervezése és egyes almoduljainak bemutatása szerepel ebben a részben, a különböző almodulok interfészei és azok egymáshoz kapcsolódása.

A harmadik részben az elért eredmények értékelése és a továbbfejlesztési lehetőségeket találhatóak.

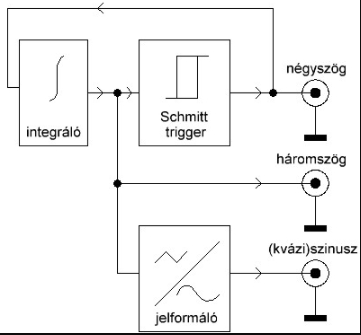
# Irodalmi áttekintés

## Függvénygenerátorok

A függvénygenerátorok nemszinuszos jelekből állítanak elő kváziszinuszos jelet. Nemszinuszos jel lehet pl. a négyszögjel, a háromszögjel, a fűrészjel. Ilyen típusú jeleket az úgynevezett relaxációs oszcillátorokkal állíthatunk elő. Belátható, hogy a szinuszos jel előállítása ebben az esetben közvetett úton történik, így ezek az áramkörök nem sorolhatók a szinuszos oszcillátorok közé, ezért részletes tárgyalásuktól eltekintve, az ide vonatkozó szakirodalom kutatását az olvasóra bízzuk. Célszerű azonban áttekinteni azt, hogyan lehet a relaxációs oszcillátor által előállított hullámalakokból (pl. négyszögjel, háromszögjel) kváziszinuszos jelet előállítani.

Az elektronikában a fentebb említett szinusz alakú rezgéseken kívül szükség van más formájú jelekre is, amelyeket relaxációs oszcillátorral állítanak elő. Például a négyszög alakú jelek (négyszögjelek) digitális technika, órák és számláló áramkörök részére, a háromszög és fűrészfog alakú jelek a katódsugárcsövek (analóg oszcilloszkópok, analóg televíziók) működéséhez szükségesek. Előállításuk nemlineáris kapcsoló áramkörökkel, például Schmitt-trigger, UJT-vel (egyátmenetű tranzisztor) lehetséges, de készülnek speciálisan jelforrás célú integrált áramkörök is. Relaxációs oszcillátorok azok az áramkörök, amelyek nem szinuszos hullámformákat állítanak elő.

Ezek az oszcillátorok egy kapcsolóeszközzel ellátott visszacsatolási hurokból állnak, amely egy kondenzátort egy ellenálláson keresztül tölt és kisüt, amíg el nem éri a küszöbértéket. Itt az oszcillátor időtartama a kondenzátor időállandójától függ. Az UJT relaxációs oszcillátorban az UJT-t használják kapcsolóként, amely feltölti és kisüti a kondenzátort.



## Jelek spektruma

A mintavételezés megértéséhez szükségünk van a spektrum fogalmának ismeretére. Periodikus jeleknél gyakran beszélünk ismétlődési frekvenciáról, annak felharmonikusairól, amplitúdó-és fázisviszonyairól. A Fourier-analízis segítségével a jelekre frekvenciatartománybeli leírás adható meg, amellyel egyszerűbben és tömörebben vizsgálhatjuk a jeleket. A jelfeldolgozás során szétválaszthatjuk az egyes frekvenciakomponenseket. Egyes rendszerek vizsgálatánál gyakran használunk periodikus gerjesztőjeleket, mivel az állandósult állapot beállása után a kimeneten is periodikus választ kapunk, így a gerjesztés és válasz arányából megkaphatjuk a rendszer átvitelét:

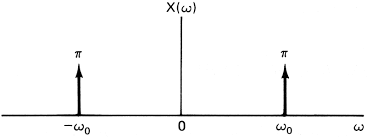
Valós értékű periodikus jel mindig felírható szinusz- és koszinuszfüggvények lineáris kombinációjaként. Ezt Fourier-sorfejtésnek hívjuk.

A periodikus jeleken kívül más jelalakokra is szükség van, ezen jelek frekvenciatartománybeli áttérésére szolgál a Fourier-transzformáció. Ennek feltétele, hogy az adott jel abszolút integrálható legyen. Ha periodikus vagy más abszolút nem integrálható jelenek szeretnénk a spektrumát előállítani, azt Dirac-delták segítségével tudjuk megtenni. A Fourier-sor által számított együtthatókat egy megfelelő nagyságú Dirac-delta segítségével tudjuk Fourier-transzformálni. Ez a következő képletből adódik:

A jel alapfrekvenciája az , a pedig a frekvencián lévő Dirac impulzus. Ezekből adódik a periodikus jelek úgynevezett „vonalas spektruma”.

Ha a spektrumot szeretnénk digitális úton feldolgozni, akkor a jelünket csak bizonyos időbeli felbontással ismerjük a mintavételezés miatt:

A képletben a mintavételi időköz, ennek reciproka a mintavételi frekvencia. Így T periódusidő alatt mintát veszünk. A levezetésből a téglalapszabály segítségével a következő összefüggést kapjuk:

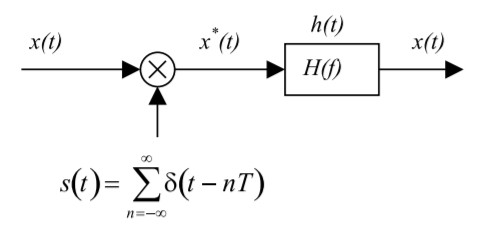
Ez a képlet a diszkrét idejű jel Fourier-sorának definíciója és az tag nélkül pedig a diszkrét Fourier-transzformált (DFT). Ebben az összefüggésben a mintavételi körfrekvencia. Mivel a valós jelek Fourier-sora és transzformáltja a negatív frekvenciákon a pozitív frekvenciák komplex konjugáltja, ezért ezen frekvenciák abszolút értéke a nulla frekvenciára szimmetrikus. Így elég csak a pozitív frekvenciákat megjeleníteni.

1. ábra szinuszjel spektruma

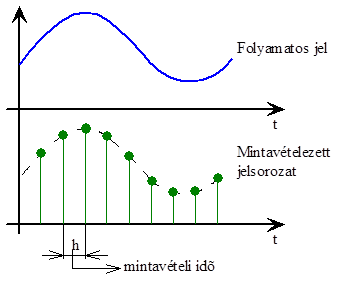
## Mintavételezés

Modern világunkban az információ feldolgozása főként digitális úton zajlik, mert így egy digitális eszközzel pontosabban és a különböző zavaró tényezőktől mentesen tudjuk az információt feldolgozni. Viszont a minket körülvevő világ főként analóg, vagyis folytonos idejű és folytonos értékű jelekből áll. Ezért szükséges van analóg jelekből diszkrét idejű és diszkrét értékkészletű jeleket, gyakorlatilag számsorozatokat előállítani. Ezt a jelek mintavételezésével tehetjük meg. A feldolgozás után sok esetben ezeket a jeleket vissza is kell állítani. Ennek neve jelrekonstrukció. Ahhoz, hogy ezt a folyamatot megértsük szükség van a jelek frekvenciatartománybeli analízisére is.

Egy jelet úgy tudunk digitalizálni, hogy egyrészt a jelből adott időközönként mintákat veszünk úgy, hogy a jel visszaállítható legyen a mintáiból. Másrészről pedig a jel értékkészletének diszkretizálását végezzük, mivel az analóg jel végtelen sok értéket vehet fel az egyes értékkészlet tartományon belül, nekünk azonban véges memória áll rendelkezésre ezen adatok tárolásához. Ezt a feladatot kvantálásnak nevezzük.

A mintavételezés modellje a következőképpen adható meg:

2. ábra Mintavételezés modellje

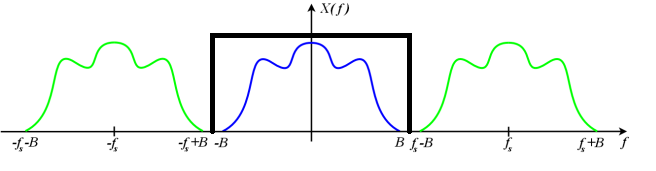
Az ábrán a bemeneten egy mintavételezni kívánt analóg jel van . Ezt a jelet összeszorozzuk (moduláljuk) egy periódusidejű Dirac-delta sorozattal. Ez annyit jelent, hogy időközönkét a jel értékét kapjuk meg a szorzás után, vagyis a jelből csak ezen időpontokban lévő értékeket látjuk a kimeneten. Így egy diszkrét idejű mintavett jelet kapunk. Ezt a következő képlettel adhatjuk meg, ahol a mintavett jel.

3. ábra Mintavételezés egy analóg jelen

Ha mintavételezett jel spektrumát szeretnénk kifejezni, akkor Fourier-transzformálni kell ezt a jelet ():

A kifejezés levezetéséből a következőt képletet kapjuk:

Látszik, hogy a diszkrét Fourier-transzformációt kaptuk meg, ami azt jelenti, hogy a mintavett jel spektruma kifejezhető a mintákkal. Ugyanakkor a mintavett jel spektrum kifejezhető a folytonos jel spektrumával is a következő módon.

A képletből belátható és a spektrumábrán (xxx ábra) is jól megfigyelhető, hogy a mintavételezés hatására a jel spektruma periodikusan ismétlődik, mégpedig a mintavételi frekvencia k-szorosával. Ez az ismétlődés a pozitív és negatív frekvenciákon egyaránt megjelenik. Tehát a jel legnagyobb frekvenciája is megjelenik a spektrum negatív szakaszán, így adódik ki a jel sávszélessége. Mindezek a következő ábrán láthatók.

4. ábra Jelek spektruma

A mintavételi frekvenciát úgy kell megválasztanunk, hogy az eredeti jel visszaállítható legyen a mintáiból. Vagyis a jelünket tudni kell rekonstruálni, amit pedig egy átviteli karakterisztikájú szűrővel tudunk megoldani, ami a xxx ábrán látszik és a matematikai megfogalmazása a következő:

egy ideális aluláteresztő szűrőt valósít meg, amely a jel B sávszélességéig enged át komponenseket, a többit szűri. Vagyis a spektrum komponenseit kiszűri.

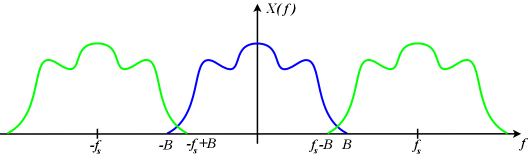
**A mintavételi tétel:**

Legyen egy folytonos idejű jel , ami abszolút integrálható. Ennek a spektruma .

Legyen ez a jel sávkorlátos határfrekvenciára.

Nyquist-Shannon mintavételi tétel kimondja, hogy egy jel visszaállítható a mintáiból, ha a mintavételi frekvencia legalább kétszerese a határfrekvenciának. Mindez formálisan kifejezve:

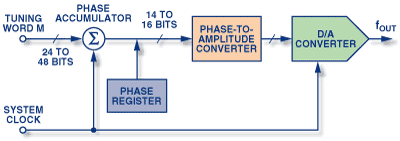
A fent említett modell a valóságban nem valósítható meg, mivel Dirac-delta generátor a valóságban nem létezik és az ideális aluláteresztő szűrő sem realizálható, mert nem kauzális a hálózat. A valódi mintavételezés egy mintavevő-tartó és egy A/D átalakítóval valósítható meg, ez látható a következő ábrán:

Átlapolódás: ha a mintavételi tétel nem teljesül, vagyis a jel sávszélessége nagyobb, mint a mintavételi frekvencia fele, akkor átlapolódás jöhet létre, másnéven aliasing. Ez azt jelenti, hogy a spektrum -el eltolt komponense „belelóg” a 0 frekvenciára szimmetrikus spektrumba. Így a jelünk nem lesz rekonstruálható.

5. ábra Átlapolódás jelensége

## Direct Digital Synthesis

A nagysebességű digitális áramkörök megjelenése lehetővé tette nagyfrekvenciás periodikus jelek közvetlen digitális előállítását. Ennek módja a Direct Digital Synthesis, röviden DDS. Lényege, hogy digitálisan hozhatunk létre különféle hullámformákat. Ennek a módszernek számos előnye van, egy ilyen például az egyszerűség. Az egész elfér egy integrált áramköri tokban és tetszőleges hullámforma létrehozható velük. Továbbá gyorsan és stabilan beállítható vele egy tetszőleges frekvencia, ami csak a referencia frekvenciától és a memóriában tárolt minták számától függ. Azonban hátrányuk is van, mégpedig, hogy a kimeneten hamis frekvenciakomponensek is megjelennek. Amennyiben ezek nagyfrekvenciás komponensek, egy aluláteresztő szűrővel könnyen kiszűrhetőek, viszont megjelenhetnek olyan nem kívánt komponensek, amik az előállítani kívánt frekvenciák közelében helyezkednek el.

A DDS működésének lényege, hogy a memória tárolja egy adott hullámforma különböző fázishelyzetéhez tartozó aktuális pillanatértékeit. A memória címbemeneteire van kötve egy számláló. Ez lényegében csak egy digitális számláló, amely minden órajelre előre számol. Az előzőekben leírtak szerint a számláló a memória címeivel van összekötve és a memória sorrendben tartalmazza az egymás után következő pillanatértékeket, ezért gyakorlatilag a számláló végiglépked a memórián. A kimeneten megjeleníti az aktuális címen tárolt értéket. Ezzel máris előáll egy digitális jel, ami gyakorlatilag egy számsorozatból áll. Ezt a műveletet fázisakkumulálásnak hívják és a Numerikusan Vezérelt Oszcillátor valósítja meg (Numerically-Controlled Oscillator). Ha ezt a digitális jelet rákötjük egy D/A átalakítóra, akkor a kimenetén a megjeleníteni kívánt analóg függvényt kapjuk. A frekvenciát be tudjuk állítani egy a bemenetére adott értékkel, amit az ábrán M jelöl. Ez az érték az úgynevezett fázislépés. Ugyanis ez az érték mondja meg, hogy hányasával lépjen végig a memória elemeken a számláló. Minél nagyobb ez a fázislépés, annál nagyobb lesz a kimeneti frekvenciánk, mivel ilyenkor gyorsabban megy végig a memórián. Azonban ennek ára van, mert a felbontás ezzel a fázislépéssel fordítottan arányos, tehát csökken, ha nő a frekvencia.

6. ábra Direct Digital Synthesis

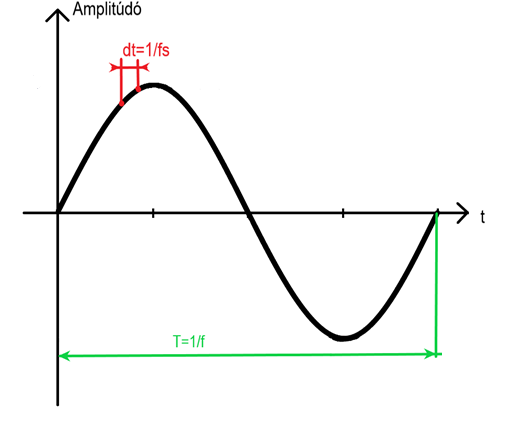
### Numerically-Controlled Oscillator (NCO)

Az NCO hozza létre a digitális hullámformát. Teszi ezt az előzőekben leírt módszerrel. Ezt az alábbi python kód jól szemlélteti. A kimeneti jel több mindentől függ:

* mintaszám
* fázislépés

Mintaszám meghatározása:

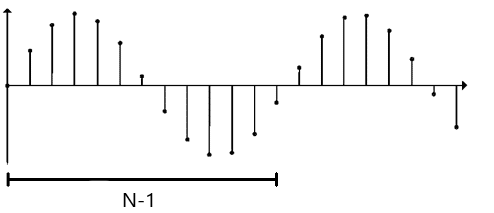
A memóriában találhatóak meg a hullámforma amplitúdóértékei, ezeket az értékeket a megfelelő függvénnyel tudjuk létrehozni és egymás után egy vektorban tárolhatjuk. Ezt a vektort ezután az FPGA memóriájába tudjuk tölteni. Ahhoz, hogy a frekvencia az elvárt tartományban mozogjon ezen értékek számát és a fázislépést is megfelelőre kell választani. A függvény a következőképpen számolható, jelen esetben egy szinusz függvényről beszélünk.

Elsőként mintákat kell előállítani a jelünkből. Ezt úgy tudjuk megtenni, hogy egy szinusz jelet mintavételezünk. A megjeleníteni kívánt jelünk legyen periódusidejű, vagyis ennek reciproka lesz a frekvencia, amit a hullámforma generátorral szeretnénk létrehozni .

7. ábra Mintavétel egy hullámformán

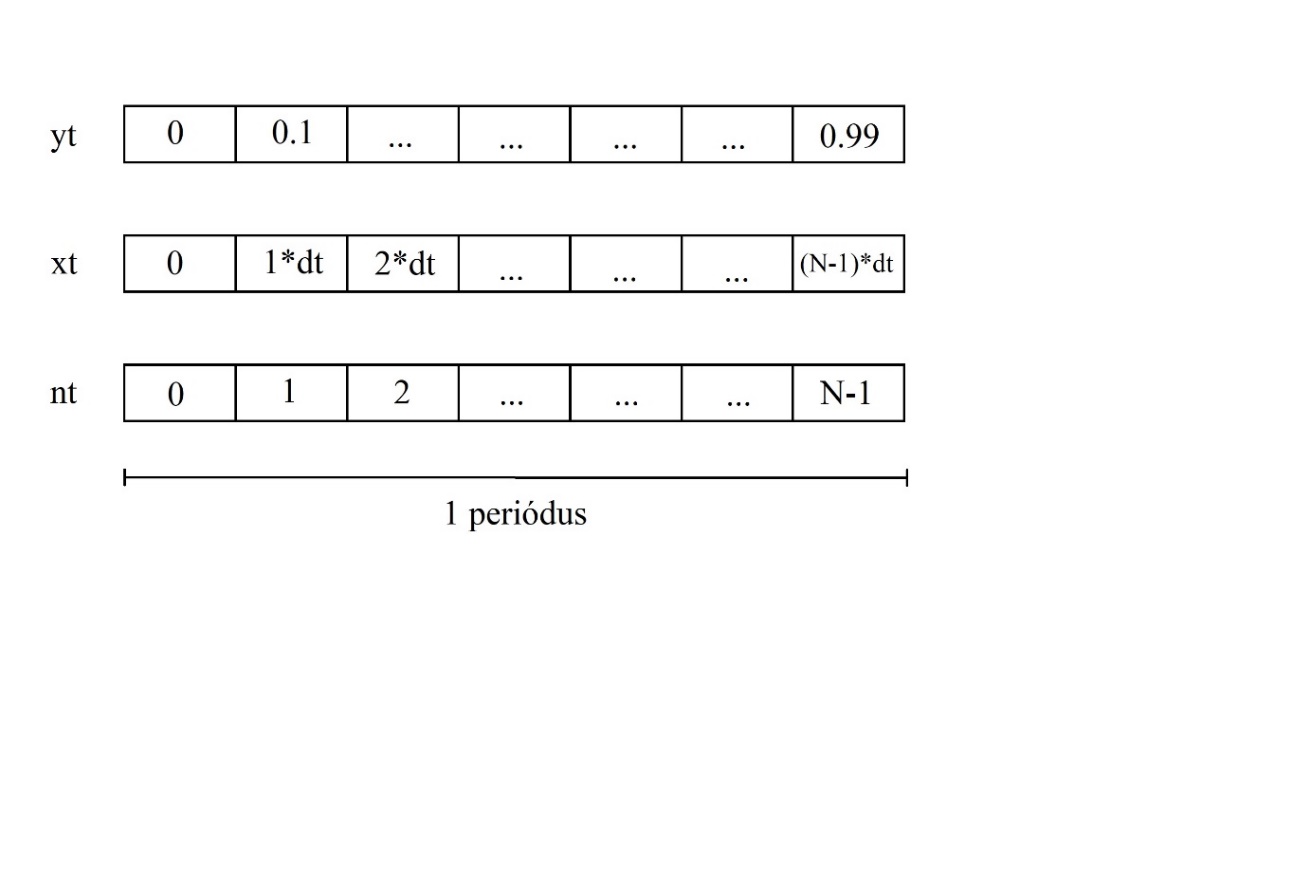
A függvény idő- és amplitúdótengelye is egy vektor. Ezeket jelölje xt és yt. Két minta között távolság van. , vagyis a mintavételi frekvencia reciproka.

Mivel az általunk előállított jel digitális és a jel 1 periódusával tesszük ezt, így meghatározható egy mintaszám (), ami a jelünk összes amplitúdóját tárolja. Ezt a következő összefüggéssel adhatjuk meg.

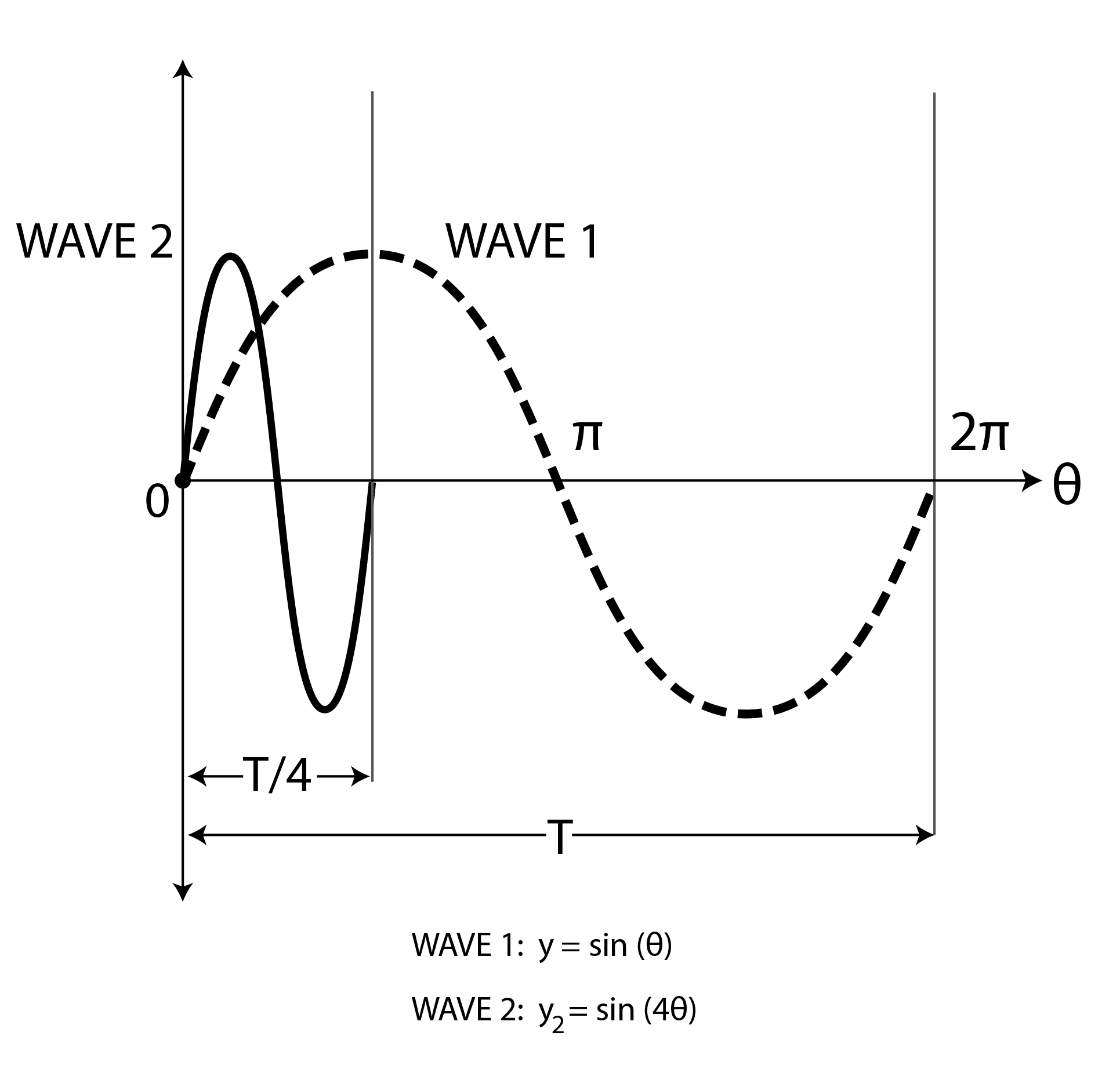
Ábrázolva a jelünk mintáit:

8. ábra Mintavett jel

Az ábrán jelölt 1 periódus mintái -edik mintáig tartanak, mivel egy periódus a 0. értéktől indul és a következő periódus is ezzel kezdődik. Egy periódusból úgy tudunk létrehozni egy periodikus jelet, hogy ezeket a periodusokat egymás után helyezzük. Ennek megvalósítása az, hogy egy végtelen ciklusba rakjuk ezt a jelet., vagyis újból a memória első értékére ugrunk után.

Miután létrehoztuk az idő vektort, amit -vel jelölünk, feltöltjük a jelünk diszkrét időértékeivel,

9. ábra Vektorok

Az vektort az ampilitúdó értékekkel töltjük fel. Tudjuk, hogy yt függvény argumentuma egy szög. Továbbá tudjuk, hogy folytonos időben egy periódus 2pi …ebből látható, hogy

10. ábra Fázisakkumulálás

Mivel , így

, ez a fázisakkumulálás. Ahol az a fázislépés

Innen megállapítható, hogy m értékének növelésével a függvény frekvenciája is megnő. Konkluzió, hogy a fázislépés változtatásával tudjuk a kimeneti frekvenciát változtatni.

Memóriaigény: az függvényértékek tárolására memóriára van szükségünk. Ez a memória N db adatot tárol. Az adatok száma függ a legkisebb megvalósítani kívánt frekvenciától.

Látszik, hogy a kimeneti frekvencia és a mintaszám fordított arányban vannak, tehát minél több mintát tárolunk el a memóriában, annál kisebb frekvenciaérték valósítható meg a hullámforma generátor által.

A legnagyobb megvalósítandó frekvencia függ a mintavételi frekvenciától. A Nyquist kritériumnak megfelelően a mintavételi frekvenciának legalább kétszer nagyobbnak kell lennie, mint a maximális kimeneti frekvencia.

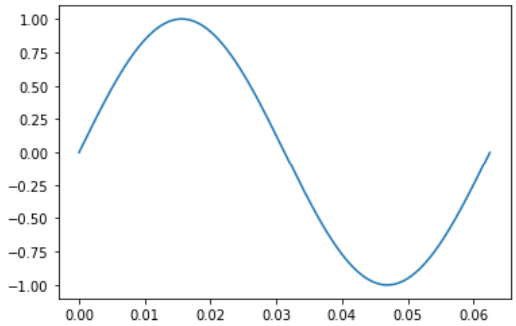
A memória igény úgy néz ki, hogy a memória mélysége és szélessége. A mélység az előző pontból kiderül, hogy a legkisebb frekvenciával függ össze. A memória szélessége az amplitúdótól és a memória

Python előállítható hullámforma:

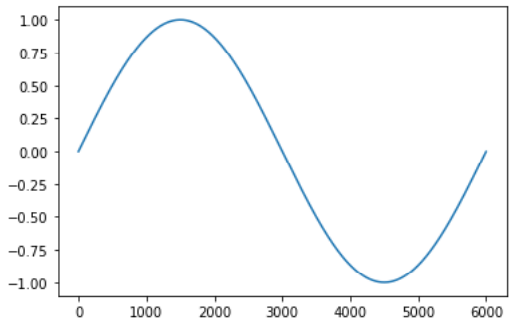
A képen szöveg látható

Automatikusan generált leírás

11. ábra NCO python kód részlet

a kódban látható, hogy létrehozunk egy N elemű vektort, ebben eltároljuk. Az xt vektor az egyes mintavételi időpontokat tárolja. Ez azt jelenti, hogy ezekben az egymástól dt távolságra lévő pontokban vesz fel értéket a függvény. Maga a függvény, az yt pedig a fázisakkumulálás révén jön létre a beállítandó frekvencia (f) és a mintavételi időpontok szerint. Ez a következő ábrán jól látszik. A függvény 1 periódusa 0.0625 másodpercig tart, ami megfelel a 16Hz frekvenciának.

12. ábra Idő szerinti függvény

A következő függvény értékei az elemek száma szerint van megjelenítve.

13. ábra Memóriacím szerinti függvény

Ebből látszik, hogy a modellünk helyes volt és az NCO-nak megfelelő viselkedést értük el vele.

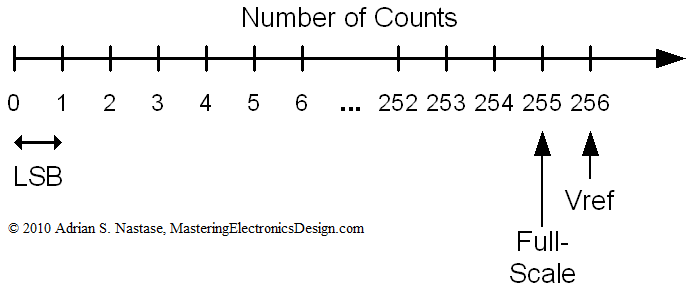
### DA átalakítás

Adott az NCO által kibocsájtott digitális hullámforma. Ezen jel a mintavételi időpontokban és egy adott kvantáltsági szinten értelmezett függvény, a mi esetünkben egy n\*1/96kHz időközönkét megjelenő 8 bites számsorozat. Ezt szeretnénk analóg jellé alakítani. Ez egy Digitális-Analóg átalakító segítségével tehető meg. Az NCO-ból kijövő jel kettes komplemens jel, mivel a hullámforma negatív és pozitív amplitúdójú is lehet (lásd python kód). A 8 bites kód értékei tehát -127 és +126 értékek között jeleníthetőek meg. Természetesen a jel bit számának növelésével finomabb felbontás érhető el.

A board-on megtalálható DAC egy 10 bites átalakító, így tehát illesztenünk kell a 8 bites jelhez. Itt az a kérdés, hogy az NCO az átalakító LSB vagy MSB bemenetéhez van igazítva, nagyobb vagy kisebb feszültséggel fog működni, a kivezérlése nagyobb vagy kisebb lesz?

Itt fontos bevezetnünk a slew rate és a full scale fogalmát.

Slew rate: jelváltási sebesség.

Full scale: maximális kivezérelhetőség

14. ábra DA átalakító illesztése

Tehát ha a kimenetet az LSB-hez igazítjuk, akkor kisebb lesz az ugrás két érték között, vagyis finomabb lesz a felbontás, de a kivezérelhetőség alacsonyabb lesz. Fordított esetben durvább felbontás mellett a kivezérelhetőség maximális lesz.

Dinamikus tartománynak (Dynamic range, DR, DNR) nevezzük azt az arányt, amelyet egy szám maximálisan és minimálisan felvehet. Legtöbbször jelek, mint hang, fény vagy feszültség/áram jelenlétében szokás említeni

Legtöbbször tízes vagy kettes alapú logaritmusát szokás venni ennek az aránynak. Más szóval a dinamikus tartomány nem más, mint a signal-to-noise (jel-zaj viszony) arányt mutatja meg nekünk. Példának okán, ha a maximális értéke a jelnek 5V, minimális érteke pedig 10 µV, akkor a dinamikus tartomány 500000:1-hez, vagy 114 dB.

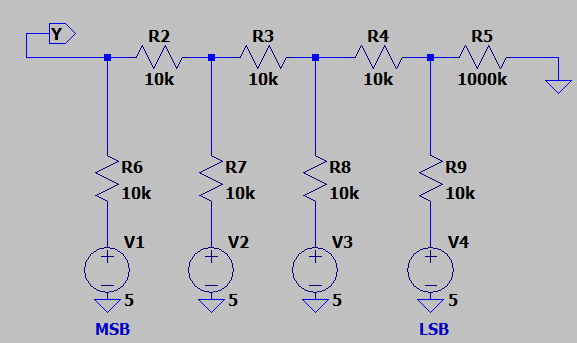


Digitális jelfeldolgozásban a dinamikus tartomány korlátozott a kvantálási hiba miatt. A legnagyobb elérhető tartomány Q bites kvantálás mellett megadható:



Példa: Ellenállás létra

DAC (Digital to Analog Converter) működésének a lényege, hogy egy digitális jelet akarunk átalakítani analóg jellé. Legyen az például egy számítógép által kibocsátott jel, mely a vezeték végén egy hangszóróba fut be. Ahhoz, hogy ezt az átalakítást véghez vigyük, a digitálisan tárolt adatot (jelen esetben a zenefájlt) át kell alakítani analóg jellé.



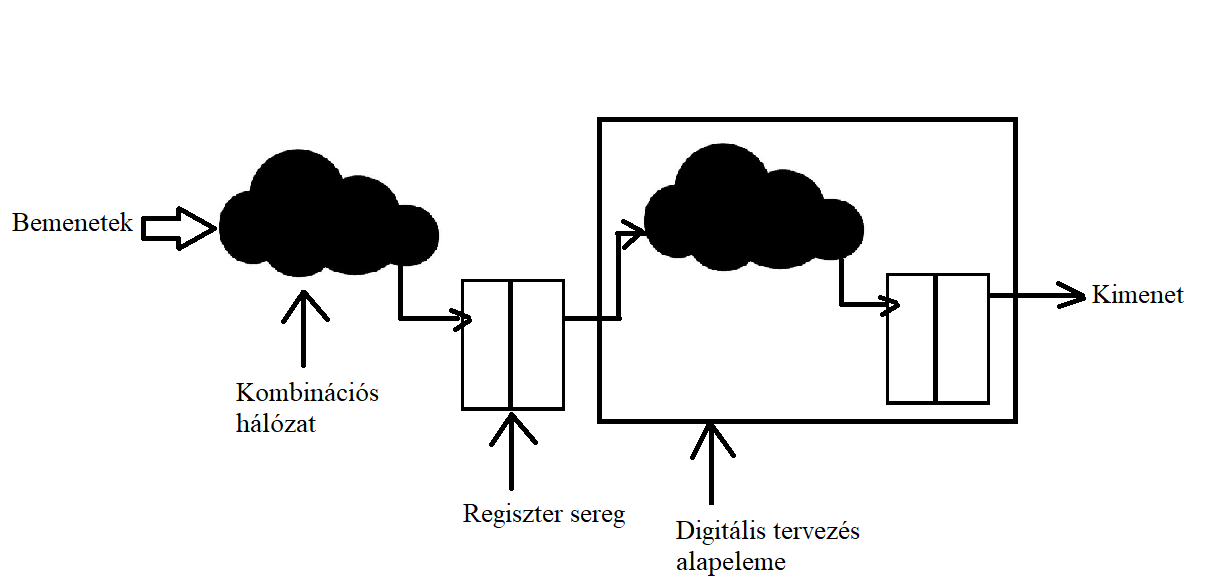
15. ábra Ellenállás létra

Ezt a DA átalakító egy adott pontossággal tudja csak megtenni. Mivel az analóg jelünk folytonos idejű, így a digitális jelekkel sohasem tudjuk teljes valójában rekonstruálni számunkra a kiadandó jelet, de tetszőlegesen meg tudja közelíteni azt. Működésének alapja a feszültségosztás. Vegyük az esetet, ha 4 bites, 0-5V DAC áll rendelkezésünkre:

Ebben a konfigurációban a feszültségosztás miatt az egyes feszültségforrások hatása az ellenállások arányában leosztódik, és különböző mértékben kerül az Y kimenetre. A kapcsolásból adódóan az LSB feszültség értéke 1/16 része lesz a maximálisan kiadható szintnek, így 5/16=312.5 mV. Ekkora lesz tehát a lépésközünk egyes “lépcsőfokok” között.

## Digitális tervezés alapjai

A digitális rendszertervezés első lépése a specifikáció. A megadott határokon belül el kell készíteni egy tervet, hogy a különböző szinten mit kell csinálnia az áramkörnek. Ezen felül az az adott részegységeket, modulokat is definiálni kell. A rendszertervezés során úgy nevezett alapelemekből építkezünk. Ezek adott mennyiségű regiszterből és az azt megelőző kombinációs hálózatból álnak. Természetesen az alább jelölt ábrán a 2. kombinációs hálózatba futhat be általunk szabályozott változó is.



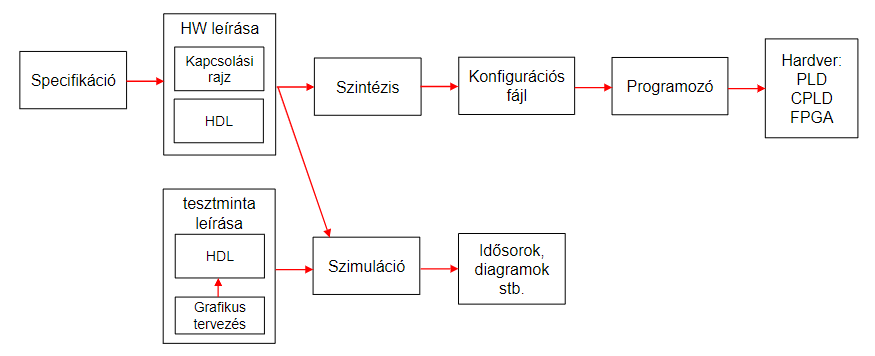
16. ábra Digitális alapegység

A digitális rendszertervezés során, kiindulva a rendszerrel szembeni követelmények megfogalmazásából, több lépésen keresztül eljutunk a rendszer fizikai megvalósításáig. A technológia fejlődése egyre bonyolultabb integrált áramkörök megvalósítását teszi lehetővé. Az összetett tervek kezelésére két módszer alakult ki áramkörök:

– leírása viselkedésük alapján

– tervezési folyamat számítógép alapú automatizálása

Egy áramkört tekinthetünk egy fekete doboznak, amelynek nem ismerjük a belső felépítését, szerkezetét, de ismerjük a viselkedését, a bemenetek és kimenetek közötti összefüggést. Egy másik megközelítés alapján ismerjük részletes felépítését, hogy modulárisan milyen alegységeket, modulokat, komponenseket tartalmaz, és a komponensek közötti kapcsolatokat. Egy áramkörnek a tervezése során meghatározhatjuk a viselkedését vagy felépíthetjük olyan áramköri elemekből, amelyek ugyanazt a viselkedést eredményezik.



17. ábra

# célkitűzések

A témalaboron kapott feladatom egy hullámforma generátor rendszertervének elkészítése és megvalósítása, amely, egy FPGA eszközön működik és egy soros porton keresztül kommunikál a számítógéppel.

A megjeleníteni kívánt jel 16Hz és 20kHz közötti frekvencia tartományban foglal el helyet. 1 db minta értékét 8 biten tároljuk. A kimeneten megjeleníteni kívánt analóg jel egy 8 bites kettes komplemens jelalak DA átalakítás után lehetséges…

A megvalósítandó funkciók a következők:

* start/stop: a rendszer elindítása és leállítása
* a frekvencia beállítása tetszőleges értékre 16Hz-20kHz-es tartományban
* DA átalakító engedélyezése, DA on/off: a jelalak megjelenítése analóg formában a kimeneten
* tetszőleges hullámforma generálása és letöltése az FPGA-ra
* hullámforma dump-olása és visszatöltése a számítógépre
* ennek megjelenítése a számítógépen és összehasonlítás a generált hullámformával
* DSP->FM moduláció megvalósítása két NCO modullal és egyéb továbbfejleszthetőségi lehetőségek

Az áramkör megvalósítása RTL szintű megtervezése, az egyes almodulok külön-külön és a toplevel integráció verifikálása. Ezután az áramkör szintézise az FPGA-ra.

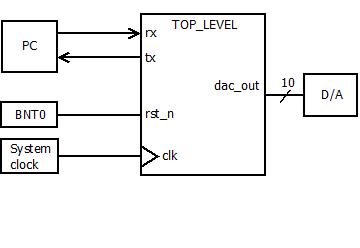
Végül a Pythonban elkészített kód által generált mintát összehasonlítom az FPGA által generált hullámformával és ez alapján vonok le következtetéseket.

# Rendszerterv

A számítógép egy Python programban generál a megadott függvényhez mintákat és ezekkel tölti fel az FPGA-ban megvalósított NCO memóriáját. A kommunikáció UART-on történik. A betöltött (jelen esetben szinusz) hullámforma mintáiból az NCO egy digitális hullámformát állít elő, amit mi felhasználunk a továbbiakban.

A következő ábrán a rendszer felépítése látható. A digitális tervezésből tanult konvekciók alapján az FPGA-ban implementálunk egy Top Level-t, amiben az egyes almodulok foglalnak helyet. Az almodulok egységbe foglalásán kívül itt valósítjuk meg a reset nyomógomb pergésmentesítését.

18. ábra Top level

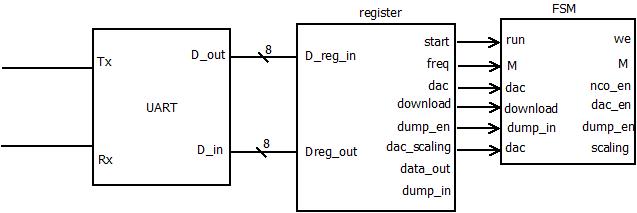


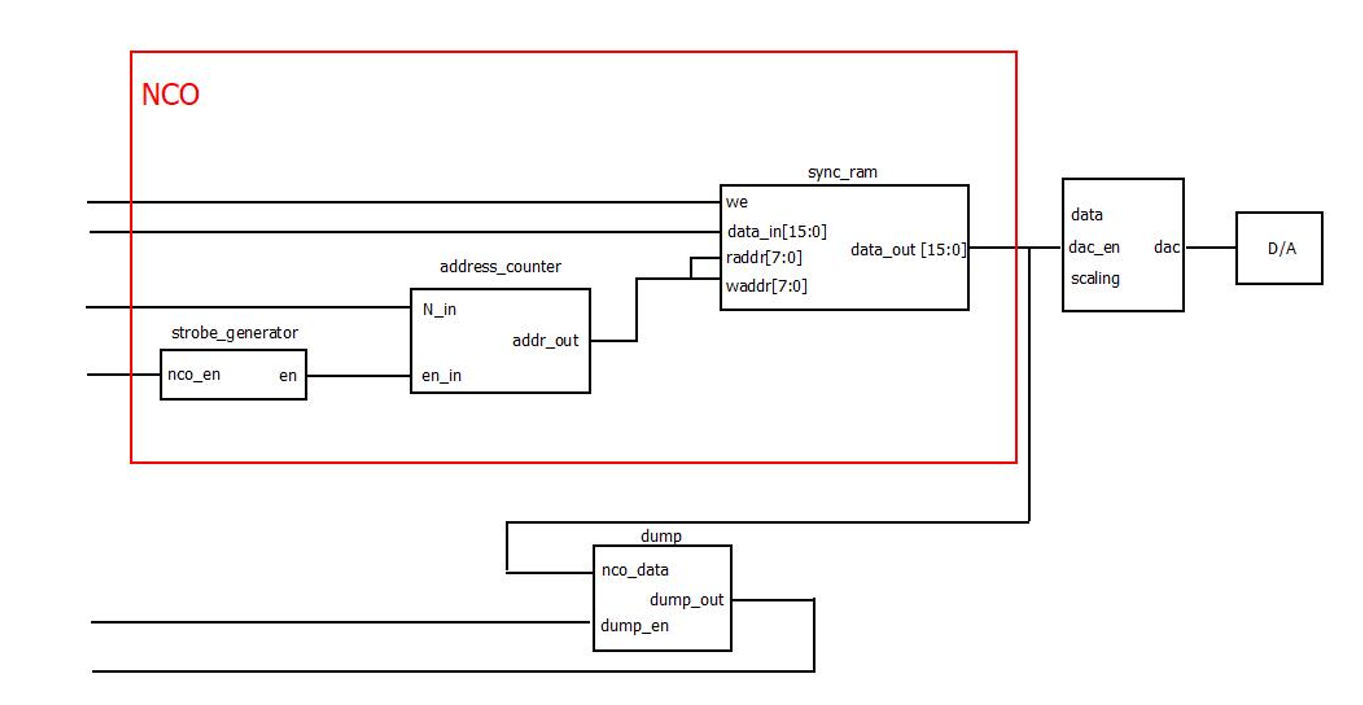
A Top Level interfészei a következők:

* tx: azaz a rendszer adatokat ezen keresztül tud küldeni a számítógépünknek,
* rx: ezt az adatok fogadására használjuk
* rst\_n: ez a reset bemenet, a reset jelet jelen esetben egy nyomógombról kapjuk
* clk: a rendszer órajele
* nco\_out: design kimenete, mely egy digitális hullámforma

A Top modulba integrálandó modellek az UART adatátviteli modul, melynek RS232 a megvalósítása. Definiálunk még egy általános regiszter fájlt, melynek a kommunikációs blokk egy interfésze. Ezután valósítjuk meg az NCO modult és két opcionális modult hozzá, a dump és a dsp modult.

A megvalósítandóáramkör blokkvázlata: kép (reg.tömb, NCO, UART, FSM, DUMP, DSP)





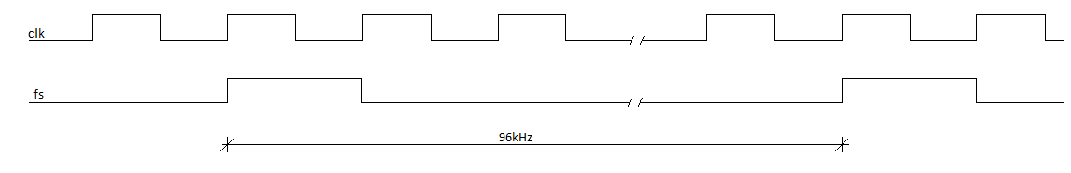
19. ábra Rendszerterv

## NCO modul

Ez a modul valósítja meg az áramkör fő funkcióját és gyakorlatilag ez generálja a megvalósítandó hullámformát. A modul a korábban már említett fázisakkumulálás elvén alapszik. Az NCO egy address regiszterből és egy memória modulból épül fel. A megfelelő nagyságú memóriába betöltünk egy a számítógép által generált függvény mintáit. Ezután az általunk meghatározott frekvencia szerinti fázislépéssel végighaladunk ezeken az adatokon, ezt végzi el az address counter. Vagyis a számláló értékét adott értékkel (m) növelve haladunk végig a számlálóval megegyező memóriacímeke. Így végül megkapjuk a kívánt jelalakot. A számlálási frekvencia a mintavételi frekvencia.

**Strobe generator:**

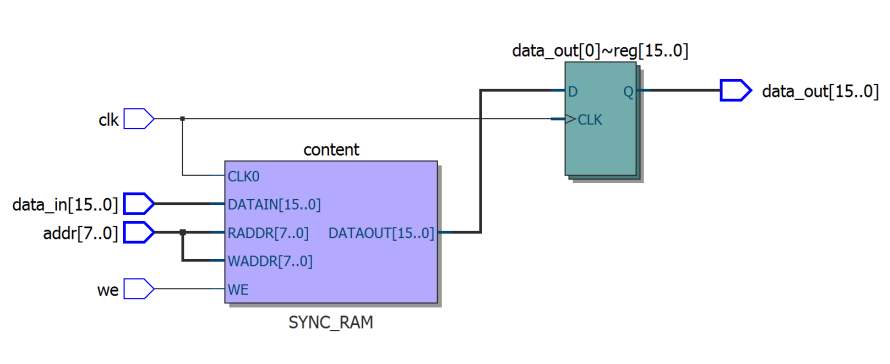
Ez a modul állítja elő az address counter számára az engedélyező jelet. Ez egy adott órajelciklusonként előállított impulzus, aminek a frekvenciája a mintavételi frekvencia, vagyis 96kHz. Ehhez a boardon található kristály oszcillátor által előállított 50MHz-es órajelfrekvenciát használjuk. Úgy állítjuk elő ezt a jelet, hogy frekvenciaosztást alkalmazunk.

Ez az jelenti, hogy minden 520-adik órajelre adunk ki egy egyetlen órajel hosszúságú impulzust.

20. ábra Engedélyező jel előállítása

**Address counter:**

Az NCO ezen része állítja elő a megjeleníteni kívánt memória elem címét. Gyakorlatilag ez egy felfelé számláló áramkör, aminek a kimenete rá van kötve a memória cím bemenetére, tehát a számláló értékének megfelelő memóriacímű érték kerül kiolvasásra a memóriából. A modulnak van még egy M bemenete is, amelyben a fázislépést adjuk meg neki és ez alapján állítja elő a címeket a memória számára.

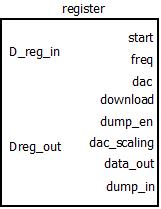
**Memória modul:**(hivatkozás a met előadásra):

**21**. ábra Syncron block ram [1]

A képen egy szinkron memória modul látható, melyet az FPGA-ba integrált block ram valósít meg. Az eszközben található 105 darab m4k nevezetű block ram, ami annyit tesz, hogy 105 4KB memóriájú modul foglal helyet a chipben. Az általunk használt memória mélységét 6 kbyte, melyben 6144 byte tárolható és nekünk 6000 elem tárolása szükséges (továbbfejlesztés dinamikus memóriával). A memóriához szükséges címvezetékek száma 13.

A memóriánk szélességét 8 bitre választottuk, tehát ekkora egy adat a memóriánkban.

## Kommunikáció

Ez az interfész teremt kapcsolatot a kommunikációs egység és az állapotgép között, vagyis azon kívül, hogy fogadja a beérkező adatokat az adatátviteli egységből leválasztó szerepet is betölt. Az átviteli módtól függetleníti a megvalósítandó rendszerünket, tehát a későbbiekben más adatátviteli egységekkel is használhatjuk, például SPI, I2C stb.

22. ábra Általános regiszter

A modul az uartba beérkező és továbbított 8 bites adatokat fogadja a D\_reg\_in bemenetén. Ezeket úgy kell elképzelni, hogy minden egyes funkcióhoz hozzárendelünk egy-egy regisztert és ezekbe a regiszterekbe írunk, és a kimenetük vezérli az állapotgépet. Így nem az állapotgépben kell a beérkező biteket szétválogatni, ezért elegendő egy-egy bit a vezérléshez.

Egy parancs 3 bájtos és a következőképpen néz ki: R/W, address0, address1, data

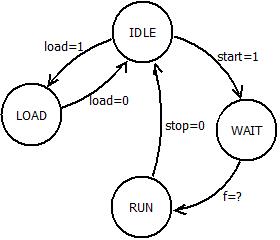
Vagyis először megmondjuk, hogy írni vagy olvasni szeretnénk, utána megadjuk a regiszter címét, amit egy multiplexerrel kiválasztunk, aztán a beírandó adatot adjuk meg.

## FSM

IDLE: alapállapot, rst\_n és stop jel hatására

WAIT: miután elindítottuk egy frekvenciaértéket vár, ami a kimeneti frekvenciája lesz az áramkörnek.

RUN: az nco modul engedélyező jelét egybe állítjuk és így működésbe lép a modul

LOAD: új hullámforma értékeket akarunk a memóriába tölteni, ha ez megtörtént visszaugrunk alap állapotba (idle)

23. ábra Állapotdiagramm

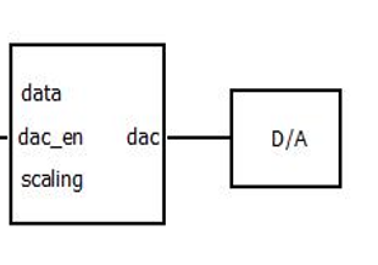
Az állapotgép állítja elő az NCO működéséhez szükséges jeleket. Segítségével tudjuk engedélyezni a dump modult és a digitális-analóg átalakítót az analóg jel előállításához.

## Dump

Ez az nco kimenetén megjelenő digitális hullámformát tölti be a memóriába és küldi tovább a számítógépnek. Ezáltal lehet a jeleinket számítógép segítségével vizsgálni. Ennek egy engedélyező bemenete van és maga az adat bemenetén jönnek a hullámforma kiolvasott adatai.

## DAC

Az nco kimenetén egy digitális jelalak jelenik meg, amit át tudunk alakítani analóg jellé egy D/A átalakítósegítségével. Ezt engedélyezhetjük és akár a be is állíthajuk hogyan kapcsolódjon a kimenetünk (8 bit) az átalakítóhoz.



24. ábra DA átalakító modul

## Reset szinkronizáció

A megvalósított áramkör minden szekvenciális elemére egy reset jel csatlakozik. Ez a jel egy alacsony aktív reset, melyet egy nyomógombbal tudunk előállítani. Azonban a nyomógombot pergésmentesíteni kell és szinkronizálni. Hogyan: az áramkör érzékel egy lefutó élt (lefutó éldetektor) számláló 1 ms-ig számol, megvalósítása szaturációs számlálóval (50 000). Minden egyes élváltásnál ez az érték lenullázódik (éldetektor). Ha a számláló eléri a beállított értéket, akkor kiad egy reset impulzust.

# Eredmények értékelése, jövőbeli fejlesztések

A Direct Digital Synthesis elvét megértettem és ennek alapján a rendszertervet is elkészítettem. Az NCO almodulját és top level modulját is elkészítettem. Az FPGA áramkört működésre bírtam a félév során és teszteltem.

Továbbfejlesztési lehetőség: a jelnek csak a negyed részét tároljuk el, ez elegendő információ ahhoz, hogy a jelet matematikai úton teljes egészében elő tudjuk állítani. Ezzel memóriát tudunk spórolni. Azonban ez csak szimmetrikus jelekre igaz például: szinusz, koszinusz stb. DSP modul továbbfejlesztése: ebben a modulban megvalósíthatunk olyan áramköröket, mint például digitális szűrés, különböző modulációkra is használható, ha az NCO- ból többet létre tudunk hozni különböző frekvenciaértékekkel. A kivezetett analóg jelünknek

# Irodalomjegyzék

Horváth Péter-Digitális rendszerek modellezése és szintézise

Elek Kálmán, Kovács Lóránt- Segédlet a Híradástechnika gyakorlatokhoz

<https://www.analog.com/en/analog-dialogue/articles/all-about-direct-digital-synthesis.html>  
<http://moodle.autolab.uni-pannon.hu/Mecha_tananyag/digitalis_jelfeldolgozas/ch09.html>  
<https://mogi.bme.hu/TAMOP/mereselmelet/ch16.html>  
<http://www.noise.inf.u-szeged.hu/Education/dip/Diplomamunka/MellarJanosZsolt.pdf>  
<https://slideplayer.com/slide/4766176/>

<https://en.wikipedia.org/wiki/Dynamic_range>

<https://en.wikipedia.org/wiki/Digital-to-analog_converter>

<https://hu.jf-parede.pt/what-is-ujt-relaxation-oscillator-circuit-diagram>