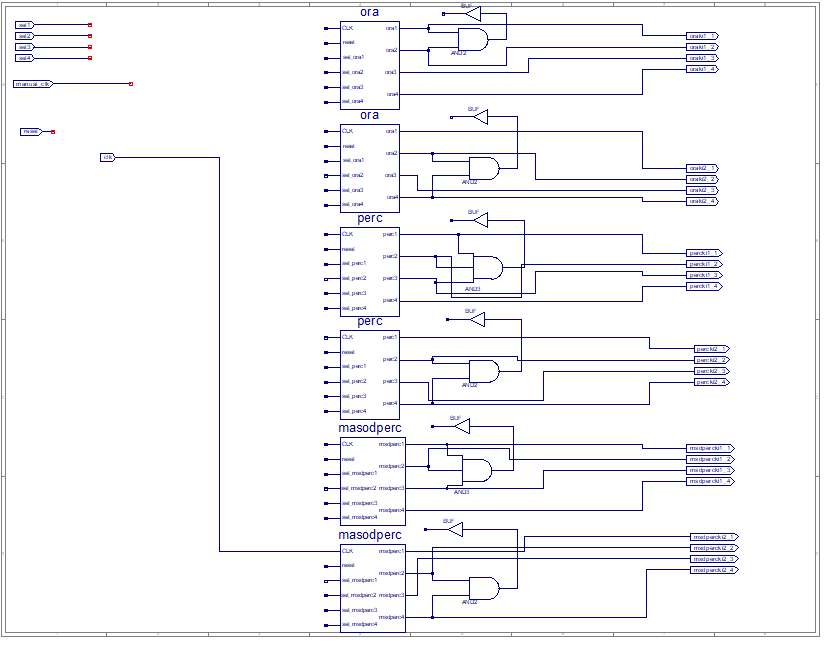
**Analóg óra elemi alkatrészekből**

**Szanyi Vilmos**

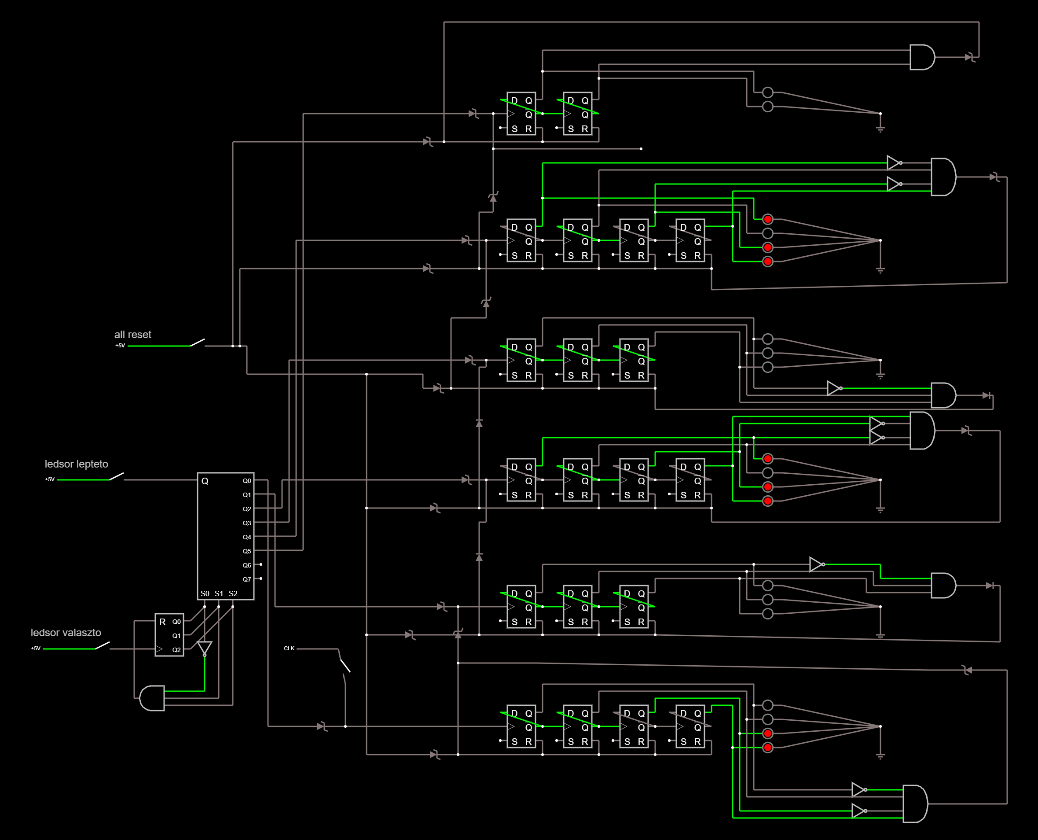
**Az elképzelés:**

Az egész projekt egy otthoni időtöltésnek indult. Apukám kérése volt, pontosabban kihívása, hogy a lehető leg elemibb alkatrészek felhasználásával készítsek neki egy olyan órát, ami ledek segítségével kettes számrendszer alapján jelzi ki a pontos időt. Én az alap elképzelés megvalósításához D-flipflopokat választottam. A koncepció nem bonyolult, végső soron 6 ledsoron kell egy felfelé számlálást végezni, amire egy up-counter is tökéletes lenne, de a kihívás miatt negált kimenetekkel sorba kötött flipflopokkal helyettesítettem. Viselkedés tekintetébe azonos a két rendszer, de mint utólag kiderült ezzel a koncepcióval később továbbfejleszthető az óra, hogy az idő pontos beállítását a flipflopok SET pin segítségével egy mikrovezérlő végezze el, de ez egy későbbi kihívás lesz. A hat ledsor nem azonos hosszúságú ugyanis az órák esetében 24-ig, a percek és a másodpercek esetében 60 ig kell elszámolni. Mindhárom részlegnél két-két oszlop van a két számjegy számára. Binárisan nézve a számokat láthatjuk, hogy nem szükséges mindenhol a negyedik hatványig egy-egy led, így a végső eloszlás 2-4—3-4—3-4 led lett oszloponként. A felfelé számlálás esetében mikor az alsóbb rendű időegység eléri maximumát az lead egy órajelet a következő oszlopnak és a RESET pineket használva lenullázza a saját flipflopjait. Így egy folyamatos óra jellegű számolást kapunk, amely egy 1 Hz es óra jele kötve pontosan 24 óra lefolyása alatt fut körbe. A koncepció hitelesítéséhez legelőször Xilinx ISE segítségével felépítettem a kapcsolást, amit leszimulálva megbizonyosodtam arról, hogy az elképzelés úgy működik ahogy annak kell. Habár a leszimulált kapcsolási rajz egy merőben kezdetleges változata volt az órának alap kezdő löketnek elegendő volt. A vizualizáció hiánya miatt következő lépésben egy valós idejű szimulációt szerettem volna létrehozni így elkezdtem áramkör szimulátorok után kutatni, amelyek ingyenesen elérhetőek, és egy pilótavizsgánál egyszerűbbek. A keresési feltételeknek minden tekintetben megfelelt a Falstadt szimulátor, amely segítségével ismételten felépítettem a kapcsolást és így újabb hibákra és tervezési hiányosságokra bukkantam, amelyek folyamatos javításával, rövid időn belül sikerült egy működőképes és több szempontból hibamentesített változatot felvázolnom. A következő lépés a végleges schematic megalkotása volt, amely értelem szerűen a leghosszabb lépése volt a teljes folyamatnak, amit az is szemléletesen mutat, hogy a mai napig tökéletesítésre és csiszolásra vár. A schematic megalkotása közben, mint ahogy az várható volt problémák tömkelegére kellett megoldást találnom. Mivel a kapcsolási rajzot mindenképpen tényleges NYÁK-ká szeretném alakítani Rögtön az Altium Designer 2018 as változatában kezdtem neki a munkának.

**Kezdeti lépések:**

****

A kapcsolás kezdetleges változata az ISE programban. Mint látható félig befejezettlen és számos egetverően súlyos hibát tartalmaz. Tekintve, hogy a Projekt legelső tehát kezdő lépése volt, leginkább az ötlet felvázolása, valamint a koncepció validálása volt a cél ennek a létrehozásával. A folyamatos tanulás elengedhetetlen volt, hogy később a már komplexebb tervezési lépéseknél a hibákat megtaláljam és kijavítsam, amelyek remélhetőleg nagyszámban meg is történtek így hamarosan végre ténylegesen elkészülhet a NYÁK.



A következő lépés volt Falstadt szimulátorban felépíteni egy már működő kapcsolást még tényleges alkatrészek kikeresése nélkül csupán a logikát vizsgálva. A program működéséből fakadóan még ez a kapcsolás is tartalmaz hibákat, de a készítése közben számos az ISE-ben vétett hiba már itt kiküszöbölésre került. A logikai alapját az órának mai napig a fent látható kapcsolás adja, amelyhez többször visszatértem ellenőrizni a készítési folyamat során. Habár a program valós időben változó szimulációt képes végezni, sokkal jobban kedveli a milliszekundumban mérhető időközöket így érthető, hogy finoman fogalmazva nem szerette ha valós 1 Hz frekvenciájú órajelet próbáltam használni egy szimulált 555ös áramkör bekötésével. A diavetítés elkerülése végett inkább egy virtuális órajelet hoztam létre, jóval gyorsabb frekvenciával így ellenőrizni tudtam a végig futás pontosságát már a ledek leolvasásával is. Itt épült meg a majd később részletezett és még mai napig finomításra váró idő beállító egység.

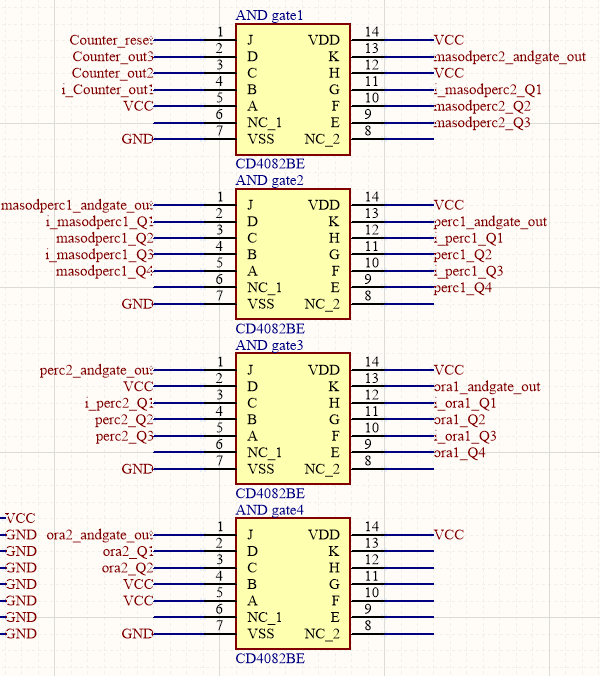
**Flipflopok:**

A képen asztal látható

Automatikusan generált leírás

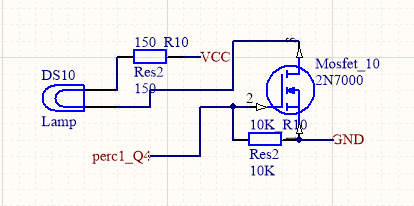
A megvalósításhoz elsőnek ki kellett választanom a megfelelő flipflopokat, amelyek a leginkább lefedik az igényeimet. Egy felfutóélre aktiválódó típust kerestem elérhető áron és hálistennek széles választékból keresgélhettem. A végső választás a CD4013-as elnevezésű kettős flipflop lett kétszer hét pines lábkiosztással. A négyszeres flipflopok használata is felmerült, viszont ezeknél a resetelés lehetősége nem volt adott. Ezek hiányában nem lehetne egyszerűen vagy beállítani, hogy egy adott oszlop számlálása mikor álljon meg, adja tovább az órajelet majd nullázódjon le. Ezért a SET és RESET pinekkel rendelkező változat lett a befutó, amely később jó döntésnek bizonyult ugyanis a későbbiekben, ha az óra fejlesztésére kerülne sor, a SET pinek segítségével egy mikrovezérlő is beállíthatja a pontos időt. Az első flipflopot leszámítva az órajelek az előző flipflop Q-negált kimenetéről érkeznek így végsősoron egy frekvenciaosztást hozunk létre a kettes számrendszer használata miatt. A kiválasztott flipflopok mind furatszerelt kivitelben lettek beszerezve a könnyebb forraszthatóság érdekében ugyanis gyakorlat híján az felületszerelt alkatrészek használatát nem mertem elvállalni. A lábkiosztás standardnak mondható: a 7 es pin a föld míg a 14 es pin a tápellátásra szolgál, így mindkét oldalon marad 6-6 pin a két flipflop számára ( kimenet; negált kimenet; órajel bemenet; RESET bemenet; D-bemenet; és a SET beállító bemenet.). Összesen 10+1 IC kerül a NYÁKra, ebből tíz látja el az óra részét az áramkörnek míg a plusz egy az 1 Hertzes órajel előállításában vesz részt. A ledeket a flipflop Q kimenetei vezérlik egy MOSFET segítségével. minden flipflophoz tartozik 1 led, ami indikálja az adott flipflopban tárolt értéket. A resetelés az adott flipflop oszlop megfelelő számértékének elérésekor a kimenetek megfelelő logikai kapcsolatának megállapításával történik 6 darab AND gate segítségével.

**Az AND kapuk:**

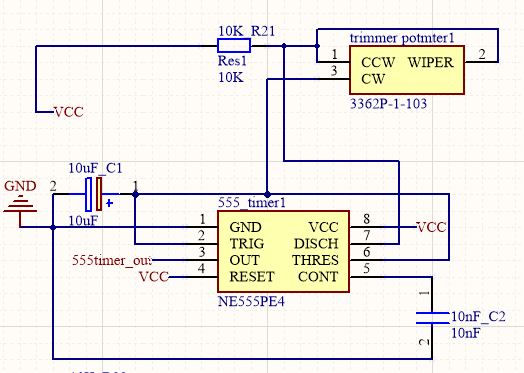


Az AND kapuk kizárólagosan a resetelés biztosításához lettek beszerelve az áramkörbe, de egy extraként az idő beállításhoz is később egy szükséges elem lesz. Jelen helyzetben négy CD4082 típusú, dupla tokozású AND gate látja el a már fentebb részletezett feladatokat. Mindegyik kapu 4 bemenettel rendelkezik, oldalankénti osztásban, tehát ha a lábakat nézzük észrevehetjük, hogy oldalanként van egy-egy be nem kötött "Non-connected" láb. A vdd pozitív tápfeszültség itt is szokásosan a tizennégyes pinre, míg a vss tehát a negatív tápfeszültség (tehát a földelés jelen esetben) a hetes lábra van bekötve. A Kapuk kimenete egyenként egy BAT85 típusú diódával van védve az esetlegesen megjelenhető káros visszáramtól. Így mikor Full Reset gombbal adunk feszültséget a Reset pineket összekötő vezetékre, nem károsodnak az AND kapujaink. Ezek a logikai kapuk így garantálni tudják a flipflopok (tehát közvetve a ledek) megfelelő nullázódását és a nem lehetséges időpontok kijelzésének megakadályozását.

**A ledek:**

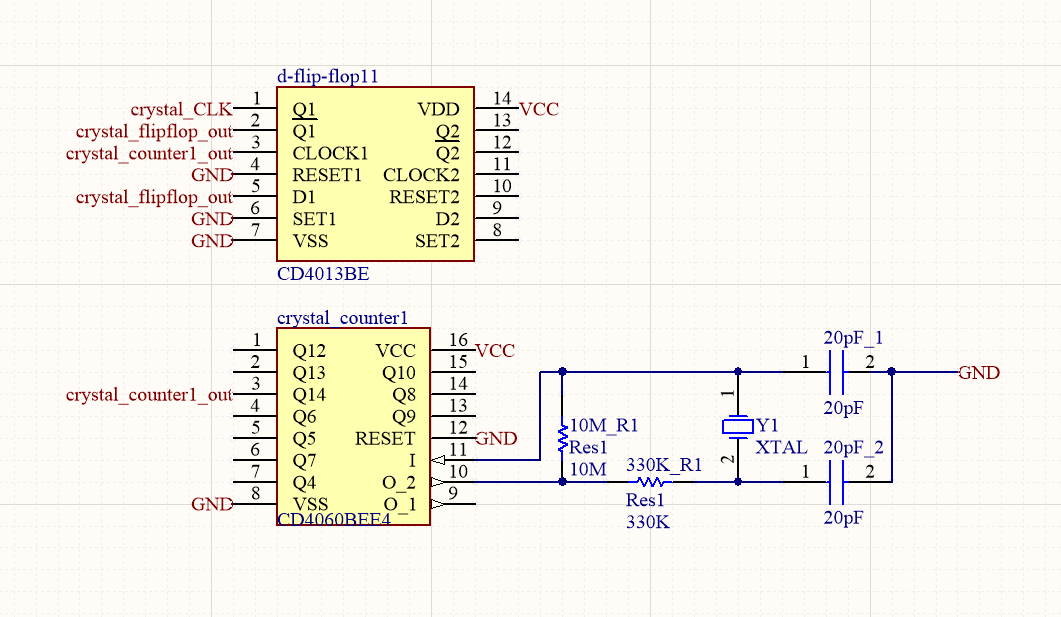


A kijelzést egyszerű Kínából nagykiszerelésben rendelt piros színű ledekkel oldottam meg. Egyenként 20 mA áramot vesznek fel és 1.9-2.1 V feszültségen működnek, éppen ezért Az 5V-os tápfeszültséget egy 150 Ohmos ellenállással csökkentettem a megfelelő szintre. Minden Led saját vezérlést kapott egy MOSFET segítségével így biztosítva, hogy a flipflopok logikai kimeneteivel tudjuk szabályozni őket. Mint a képen is látható a 2N7000 típusú N-Mosfetet használtam, amelynek a gate lábára kötött flipflop kimenet szabályozza, hogy a drain lábra kötött led negatív kivezetése kapcsolva legyen-e a source lábra kötött földeléssel. Lehúzó ellenállásként a gate és source közé be lett kötve egy 10 K Ohmos ellenállás is. Ebből az összeállításból összesen 20 darab helyezkedik el a leendő NYÁK-on. Egymás alatt helyezkednek el hat oszlopban így reprezentálva az órákat (huszonnégyig számolva kettő és négy bitet használva), a perceket (hatvanig számolva három és négy bitet használva) valamint a perceket ( szintén hatvanig számolva szintén három és négy bitet használva) tehát minden egyes bitet egy ledfény reprezentál.

**Az 555ös órajel:**

Kezdetben pár konzultációt követően az áramkör "egyszerűsége" érdekében úgy döntöttem, hogy az óra működéséhez elengedhetetlen 1 Hertzes órajelet egy 555-ös IC vel valósítom meg. Főleg az internet segítségével megtaláltam a megfelelő kapcsolást a cél eléréséhez. A GND láb természetesen a földre van kapcsolva közvetlenül, a trigger és a treshold láb, a trimmer poti (állítható ellenállás) -val kiegészítve, a nagyobb pontosság érdekében egy 10 uF -os polarizált tantál kondenzátor közbeiktatásával, míg a control pin csupán egy 10 nF-os tantál kondenzátor közbeiktatásával van földelve. A tápfeszültség itt is 5V a vcc bemenetre kötve. Az áramkör frekvenciájának beállításához egy többfordolatú 100K ohmos ellenállás van bekötve a discharge és treshold pinek közé. Több számítást követően arra jutottam, hogy ha ez a trimmer potméter pontosan 67.1 kOhm ellenállásra van beállítva abban az esetben a kimeneten egy majdnem tökéletes 1 Hz frekvenciájú négyszög jelet kapok, amit szimuláció és mérés is igazolt a későbbiekben. Ezt a négyszög jelet egy Mosfetre kötve hoztam volna létre az első flipflop számára szükséges clk órajelet. Sajnos azonban a kondenzátorra hagyatkozás miatt közel nem érhető el ezzel a kapcsolással óra pontosságú órajel, amit már a tervezéskor észrevettem majd később Arduinoval történő mérések meg is erősítettek ugyanis másodpercenként bőven +/- 15 milliszekundumos eltérés volt megfigyelhető, ami rövid idő alatt összeadódik így pontatlanná és használhatatlanná téve az órát. Az áramkör szépsége miatt meg a tudás megszerzésének "emléket állítva" nem vettem ki a NYÁK tervéből, hanem szimplán egy kapcsolóval választható opcióként otthagytam a sarokban, hátha néha átveszi a feladatot a következő pontban részletezett kristály oszcillátortól.

**A kristályoszcillátoros órajel:**

****

Az 555-ös áramkör pontatlansága miatt hamar felmerült egy alternatíva keresése és mi se lenne jobb választás, mint a nevüket is az alkatrészről kapó kvartz órákban használt módszer. A kristályoszcillátor használata ismeretlen terep volt számomra, de jelentős utána olvasást követően belevágtam a tervezésbe. A kapcsolás lelke a már említett kvartz kristály, amely pontosan 32768 Hz frekvenciával "rezeg" (tehát oszcillál) és mi ezt a javunkra tudjuk fordítani ugyanis ez egy az egyben megegyezik a kettő 15. hatványával. Ezt kihasználva egy erre a célra gyártott CD4060 típusú 14bites számlálót vettem, amelynek, ha a tízes és tizenegyes lábára a megfelelő módon bekötjük a kristályt, a hármas lábon egy tűpontos 2Hz frekvenciájú négyszög jelet kapunk. Ezt kötjük rá a már említett plusz egyedik flipflopra, hogy egy végső frekvencia felezést követően megszülessen a számunkra szükséges 1Hz es órajel. A pontos bekötést takarja a kristály két lába egy-egy 20 pF -os kondenzátorral való földre kötése, a 10 pin elé berakott 330K ohmos ellenállás bekötése, valamint a két oszcillátor láb összekötése egy 10M ohmos ellenállással. A végső pontosság miatt ajánlani szokták az egyik kondi trimmer kondira való cserélését, de esetemben nem jelentett volna akkora pontosság növekedést, mint amennyivel megnövelte volna az alkatrészköltséget. A pontosságot alátámasztotta a már említett Arduinoval végzett hosszútávú frekvencia mérés eredménye, amely szerint a kristály +/- 1 milliszekundumos eltéréssel operált, amely egyrészt tekinthető akár mérési hibának is, másrészt a negatív változás miatt napokban mérhető távolságban gyűlhet össze 1 másodpercnyi csúszás. Ezt a mértéket már tolerálhatónak ítéltem meg, tekintve, hogy az óra főleg tanulási céllal készül, másrészt nem volt cél a teljes pontosság hiszen ezt egy hegyező méretű szimpla RTC modul sokkal ügyesebben megteheti helyettünk.

időbeállítás