

3. Digitális rendszertervezés - Szintézis FPGA technológiára

Szerző: Horváth Péter

Az IT eszközök technológiája c. tárgy laboratóriumi gyakorlataihoz módosította: Ress Sándor

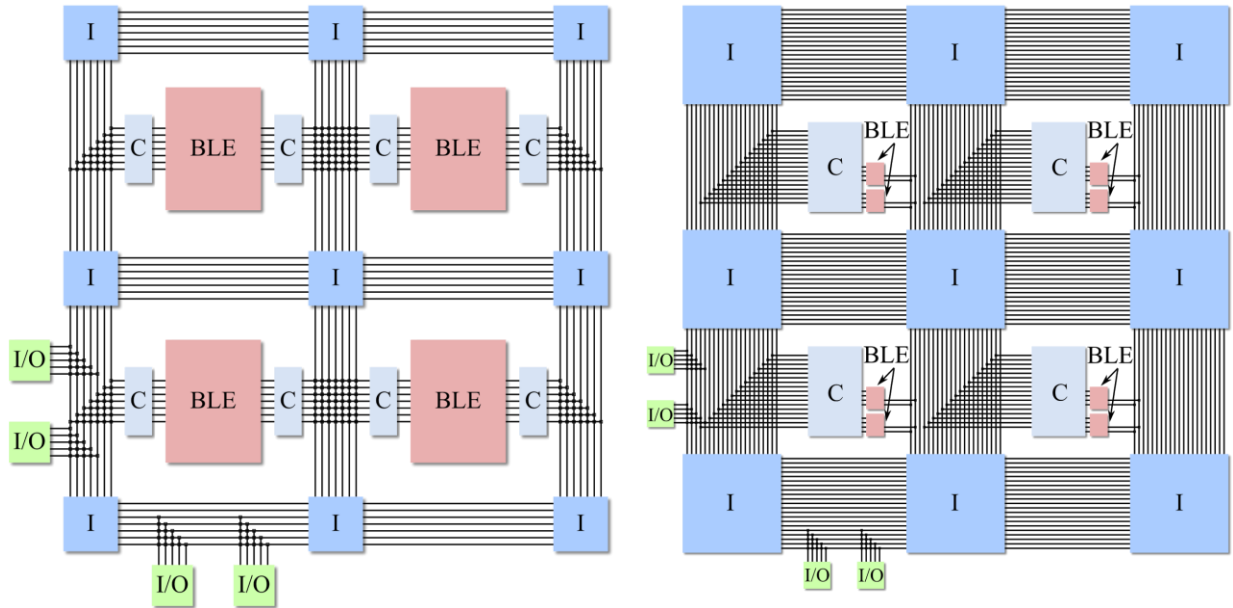
A Szintézis FPGA technológiára c. laboratóriumi gyakorlat során nagymértékben támaszkodunk az előadáson megismert logikai szintézisről tanultakra. A logikai szintézis eljárás különböző megvalósítási technológiákra nagyrészt megegyezik, tulajdonképpen nincs is különbség közöttük egészen a generikus kapusintű modellt a megvalósítás alapjául szolgáló technológia erőforrásaira leképező mapping-ig. Ettől a lépéstől kezdve azonban a két folyamat jócskán eltér egymástól. Az eltérés oka a standard cellás ASIC és az FPGA (*Field Programmable Gate Array*) áramkörök architektúrájának különbözőségében rejlik.

3.1. Az FPGA áramkörök architektúrája

Az FPGA-k olyan, **általános célú, újrakonfigurálható** eszközök, amelyek programozható erőforrásaik segítségével tetszőleges digitális funkciót képesek megvalósítani. A rendelkezésre álló erőforrások természetesen eszközzől eszközre változnak, de az alábbi három típus mindegyik FPGA-ban megtalálható:

- **konfigurálható logikai blokkok** (*Basic Logic Element, BLE*): Kombinációs és szekvenciális hálózatok alapelemeit tartalmazó funkcionális egységek.
- **konfigurálható I/O blokkok**: A külvilággal való összeköttetést biztosító funkcionális egységek.
- **konfigurálható huzalozási erőforrások**: A logikai blokkok és I/O blokkok összekapcsolását megvalósító elemek.

Az FPGA konfigurálása alatt ezeknek az erőforrásoknak az elvárt logikai funkció alapján való beállítását értjük. A 3-1. ábrán az FPGA áramkörök felépítése figyelhető meg. A bal oldali ábrán jól elkülöníthetők a különböző funkcionális egységek. Az I-vel (*Interconnect block*) jelölt blokkok az egyes huzalozási csatorna-szakaszok összeköttetéseit valósítják meg, míg a C-vel (*Connection block*) jelölt blokkok a BLE-eket kapcsolják a huzalozási csatornákhöz. A jobb oldali ábra az egyes erőforrások egymáshoz viszonyított fizikai méretét szemlélteti. Egy FPGA áramkörben a rugalmas konfigurálhatóságot biztosító erőforrások a chipnek mintegy 90-95%-át foglalják el.



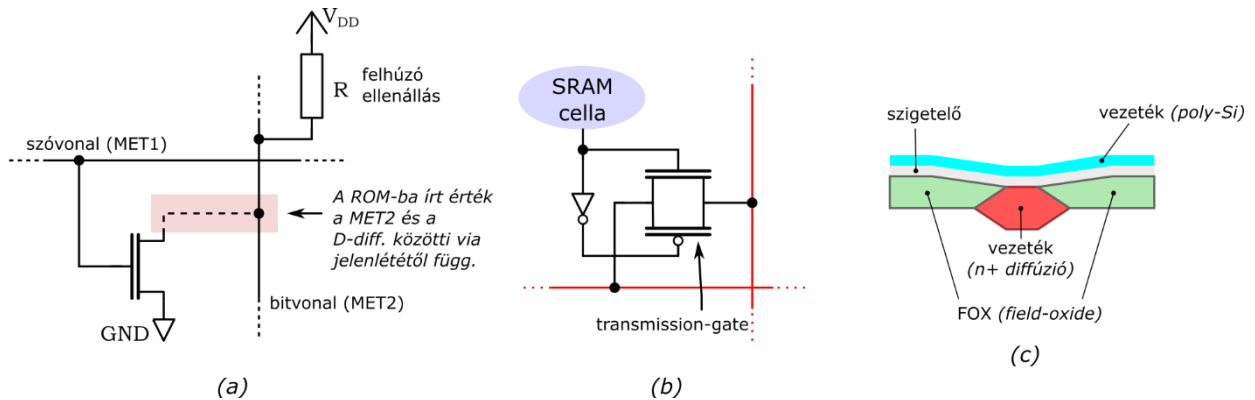
3-1. ábra Az FPGA áramkörök felépítése

A különböző konfigurálható erőforrások állapotát a működés során tárolni kell. Erre szolgál az ún. **konfigurációs memória**, amelynek fizikai megvalósítása többféle lehet. A legelterjedtebb megoldás az SRAM típusú memória, amelynek legfőbb jellegzetessége, hogy illékony, vagyis tartalmát kikapcsolás után nem őrzi meg. Ha a konfigurációt a tápfeszültség elvétele után is meg kell őrizni, EEPROM, FLASH vagy antifuse-alapú konfigurációs memóriára van szükség. Ezek a megoldások az SRAM-alapúnál drágábbak, de bizonyos esetekben alkalmazásuk elkerülhetetlen.

3.1.1. Konfigurálható huzalozási erőforrások

3.1.1.1. Konfigurálható kapcsolók

Minden konfigurálható technológia alapja a két vezeték között megvalósított, konfigurálható összeköttetés. Technológiailag a probléma sokféleképpen kezelhető, a legfontosabb megoldásokat a 3-2. ábra mutatja be.



3-2. ábra Konfigurálható összeköttetések megvalósítási lehetőségei: maszk-programozás (a), transmission-gate (b), anti-fuse (c)

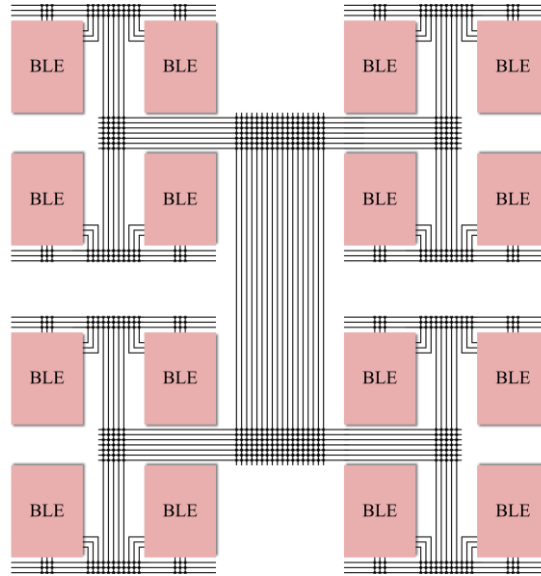
Az ún. **maszkprogramozás** a konfigurálhatóság legkorlátozottabb formája. A layout-ot úgy alakítjuk ki, hogy vezetékek közötti **összeköttetés** meglétét/hiányát csupán **egyetlen maszk határozza meg**. Ilyenkor az összes technológiai lépés előre elvégezhető, majd egyetlen maszk testreszabásával a teljes áramkör egyedivé tehető. A megoldás nem túl rugalmas, hiszen egyrészt a testreszabást nem közvetlenül a felhasználó, hanem a gyártó végzi, továbbá a legyártott eszköz konfigurációja nem módosítható.

Az ún. **antifuse** technológia esetén a konfigurálható kapcsolat két vezeték között úgy alakul ki, hogy két vezető réteg közé (pl. a 3-2. ábrán látható példában poly-Si és n+ diffúzió) vékony szigetelő réteget helyezünk el, amely a két vezető réteg közé kapcsolt **megfelelő potenciálkülönbség hatására átüt**, egy maradandó vezető csatornát képezve (innen az elnevezés, "anti-biztosíték"). A módszer előnye, hogy az összeköttetések létrehozása a gyártás után, a felhasználó által történik, hátránya, hogy a maszkprogramozáshoz hasonlóan a konfiguráció ez esetben sem módosítható.

Rugalmasság szempontjából az ún. **transmission-gate-alapú kapcsoló** technológia a legelnyösebb. Ebben az esetben a 3-2. ábrának megfelelően összekapcsolt p- és n-csatornás **MOS tranzisztorok csatornái valósítják meg az összeköttetést** a két vezeték között. A módszer előnye, hogy a transmission-gate vezérlő feszültsége tetszőlegesen módosítható. Az összeköttetés maradandósága tehát attól függ, hogy a vezérlő bitet milyen típusú memóriában tároljuk. A 3-2. ábrán látható megoldásban SRAM cellát használtunk, amely kikapcsoláskor elveszíti tartalmát, tehát az összeköttetések sem maradandók. Ha a vezérlő bitet EEPROM vagy FLASH memóriában tároljuk, akkor az ezzel a megoldással megvalósított összeköttetések a tápfeszültség elvételekor is megmaradnak.

3.1.1.2. Globális huzalozási architektúra

A huzalozási csatornák és logikai blokkok egymáshoz viszonyított helyzete alapján **hierarchikus és sziget típusú** architektúrákat különböztetünk meg. A 3-3. ábrán ez utóbbira láthatunk példát. A hierarchikus huzalozási architektúrát a 3-3. ábra mutatja be. Ezen az ábrán a jobb áttekinthetőség céljából kihagytuk a huzalozási csatornák és logikai blokkok összekapcsolását megvalósító erőforrásokat.

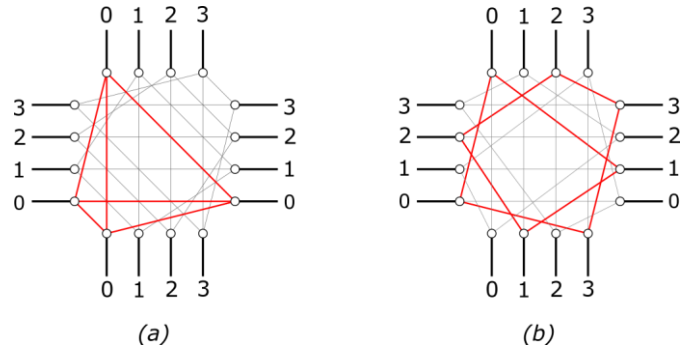


3-3. ábra A hierarchikus huzalozási architektúra sémája

A **hierarchikus** huzalozási architektúra előnye a sziget típusúval szemben, hogy a **vezetékek késleltetése jobban prediktálható**, ugyanakkor a logikai modell erőforrásainak leképezése az eszköz erőforrásaira problémás lehet, ha a logikai hierarchia nem követi a huzalozás hierarchiáját, és a huzalozási hierarchia által meghatározott logikai blokk-csoportok mérete nincs összhangban a logikai terv blokkjainak erőforrásigényével. A hierarchikus vezetékezésben az áttérés az egyik logikai blokk-csoportból egy másikba jelentős késleltetésnövekedést jelent. E problémák miatt **a mai modern FPGA-k inkább a sziget típusú globális huzalozási architektúrát alkalmazzák**. A sziget típusú huzalozási architektúrában - ahogy az a 3-1. ábrán is látható - a logikai blokkok egy kétdimenziós mátrix rácspontjaiban helyezkednek el. A közöttük futó huzalozási csatornáknak különböző hosszúságú vezeték-szegmensek és az azokat összekapcsoló kapcsolómátrixok helyezkednek el.

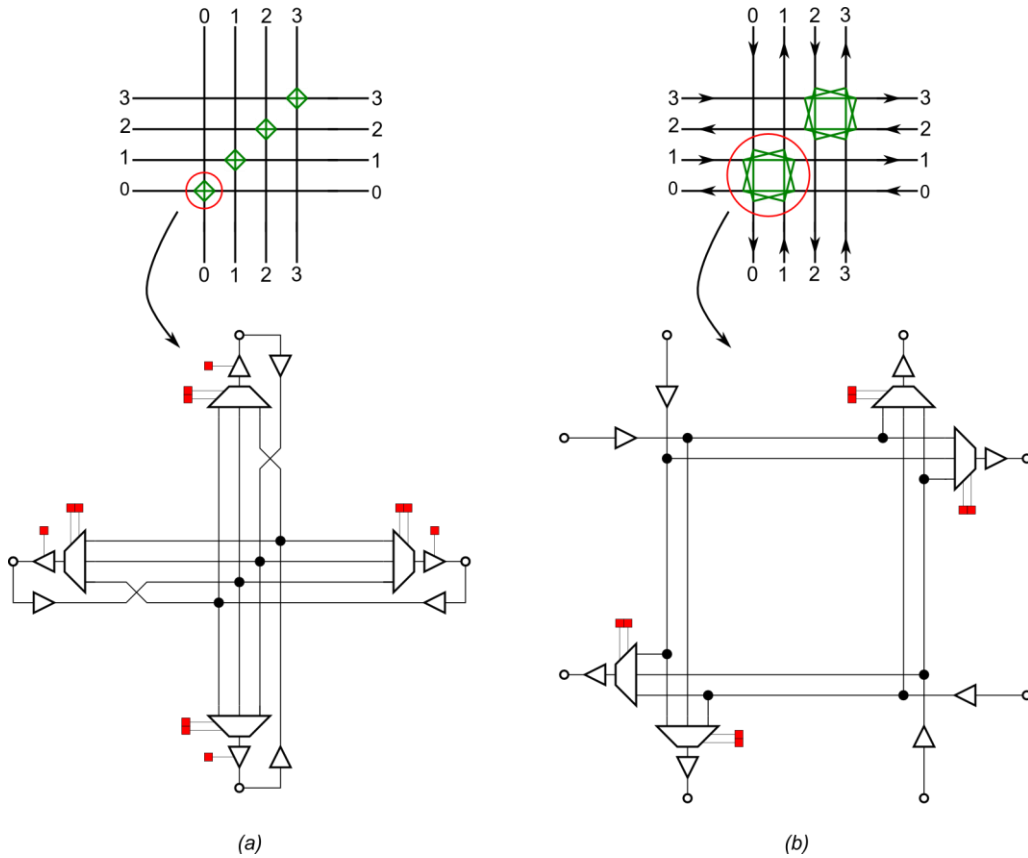
3.1.1.3. Kapcsolómátrixok

A 3-1. ábrán I-vel jelölt kapcsolómátrixok a lehetséges összeköttetéseknek csak egy részét képesek megvalósítani, mivel a tetszőleges vezetékpár összekapcsolására képes szerkezet erőforrásigénye megengedhetetlenül nagy lenne. A 3-4. ábra a *Disjoint* és a *Wilton* típusú kapcsolómátrixok által megvalósított összeköttetéseket mutatja be.



3-4. ábra Disjoint (a) és Wilton (b) típusú kapcsolómátrixok

A kapcsolómátrixokon belüli összeköttetéseket irányuk alapján is megkülönböztethetjük. A 3-5. ábrán látható a kétirányú és a csak egyirányú összeköttetések megvalósítására alkalmas, multiplexer-alapú kapcsolómátrixok egyszerűsített kapcsolási sémája.

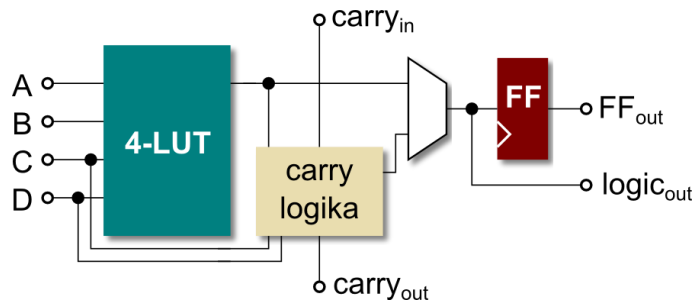


3-5. ábra Kétirányú (a) és egyirányú (b) összeköttetések a kapcsolómátrixon belül. A piros négyzetek a konfigurációs memóriabeli biteket szimbolizálják

A BLE-k és a vezetékszegmensek közötti összeköttetéseket ugyancsak konfigurálható, bemeneti és kimeneti kapcsolóblokkok valósítják meg.

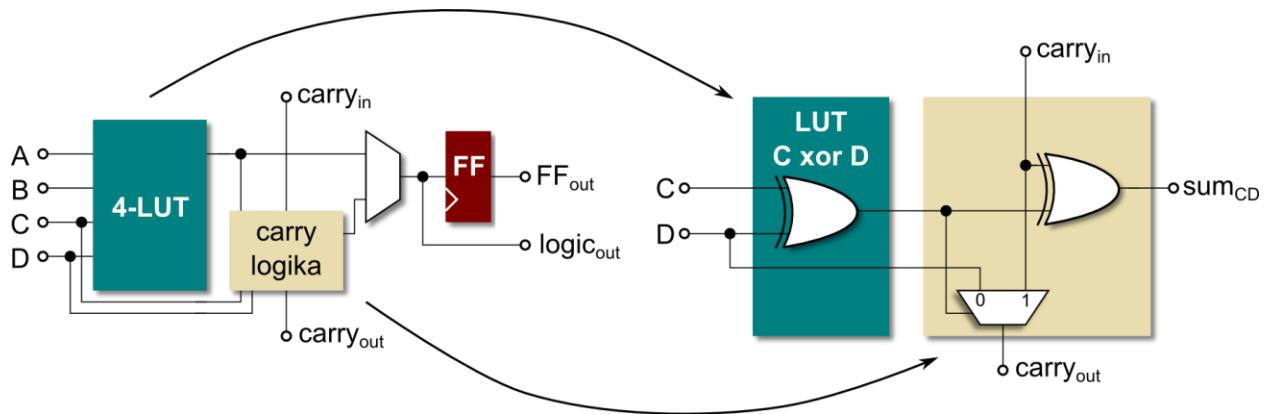
3.1.2. Konfigurálható logikai blokkok

A konfigurálható logikai blokkok a digitális áramkörök alapelemeit, **kombinációs hálózatok felépítéséhez szükséges logikai kapukat és tároló elemeket, flip-flopokat** tartalmaznak. Azt a tulajdonságot, hogy egy FPGA logikai blokkjai mennyire összetett funkciót képesek önmagukban megvalósítani (pl. egy logikai blokk csak egyetlen logikai kaput tartalmaz, vagy egy teljes számlálót, esetleg több-bites aritmetikai/logikai áramköröket), az **architektúra granularitásának** nevezzük. A mai modern FPGA-k **finom granularitásúak**, ami azt jelenti, hogy egy logikai blokk önmagában kevés funkciót lát el, azonban e kisméretű blokkokból a teljes eszköz nagyon sokat tartalmaz. A 3-6. ábra egy logikai blokk leegyszerűsített kapcsolási sémáját mutatja be. Bár a valós eszközökben a logikai blokkok lényegesen összetettebbek, az ábra jól szemlélteti, hogy melyek azok az erőforrástípusok, amelyek egyetlen konfigurálható logikai blokkból sem hiányozhatnak.



3-6. ábra Egy konfigurálható logikai blokk egyszerűsített kapcsolási sémája

Az ábrán 4-LUT-ként jelölt áramköri elem a kombinációs logikák megvalósítására szolgáló 16-bites memória. **Tetszőleges négy, vagy annál kevesebb bemenetű logikai függvény megvalósítható vele** (a bemenetek száma eszközről eszközre változik). A **carry-logika** néhány kiegészítő kombinációs hálózatot tartalmaz, amelyek az **összeadó áramkörök** megvalósításának **hatékonyságát** növelik. A 3-7. ábrán látható, miként alakítható minden egyes BLE egy-egy teljes összeadóvá a megfelelő carry-logika segítségével.



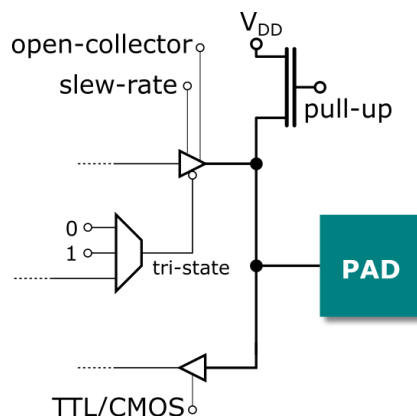
3-7. ábra A carry-logika feladata az összeadók kritikus részáramköreinek hatékony megvalósítása

Minden BLE tartalmaz továbbá egy **konfigurálható 1-bites kimeneti regisztert**, amely a LUT vagy a carry-logika kimenetét képes tárolni.

A konfigurálható logikai blokkokra a 3-1. ábrán a BLE (*Basic Logic Element*) elnevezést használtuk. Ezeket az elemeket **általános célú FPGA erőforrásnak** is nevezik (a speciális célú erőforrásokat lásd: 3.1.4. pont). A különböző gyártók más-más mozaikszót használnak saját termékeik összetevőinek megnevezésére. Ez megnehezíti a különböző gyártóktól származó eszközökre szintetizált áramkörök erőforrásigényének összehasonlítását. Tovább bonyolítja a helyzetet, hogy a BLE-eket az egyes gyártók más-más hierarchiába szervezik és az egyes hierarchiaszinteket is másképpen nevezik el. A Xilinx LC-nek (*Logic Cell*) nevezi a legalsó hierarchiaszintet, ami kb. a 3-6. ábrán látható összetettséggű áramkörnek felel meg. Két LC alkot egy slice-ot, 4 slice pedig egy CLB-t (*Configurable Logic Block*). Az Altera LE-nek (*Logic Element*) nevezi a legalsó szintet, amelyből (pl. a Cyclone eszközcsalád esetén) 16 db alkot egy LAB-et (*Logic Array Block*).

3.1.3. Konfigurálható I/O blokkok

Az FPGA áramkörök általános célú eszközök, ami azt jelenti, hogy sokféle áramkörü környezetbe beilleszthetőnek kell lenniük. Ezt a rugalmasságot a konfigurálható ki- és bemeneti blokkok, az ún. I/O blokkok biztosítják. A 3-8. ábrán egy I/O blokk egyszerűsített sémája látható.



3-8. ábra Egy I/O blokk egyszerűsített kapcsolási sémája

Ezek a blokkok határozzák meg az FPGA általános célú kivezetéseinek tulajdonságait, pl. **irányát**, kimenet esetén annak **típusát** (tri-state, open-collector), a **jelszintet** és az esetlegesen **differenciális jelvezetést**, amelyekkel az eszköz képes alkalmazkodni a különböző kapcsolástechnikával készült beágyazó környezetek igényeihez (TTL, CMOS, ECL stb.). Ha az FPGA zajra érzékeny környezetbe kerül, szükség lehet a kimenetek fel- és lefutási sebességének korlátozására a környező áramkörü komponensekbe indukált zaj mértékének csökkentése céljából. Erre szolgálnak az ún. **slew-rate control** áramkörök, amelyek ugyancsak részét képezik az I/O blokkoknak.

3.1.4. Speciális célú erőforrások

Az általános célú FPGA erőforrásokból tetszőleges digitális funkció kialakítható, azonban egyes, **gyakran alkalmazott konstrukciók nem valósíthatók meg hatékonyan**. A probléma kiküszöbölésére a mai modern FPGA-k ún. **heterogén architekturális elemeket** is tartalmaznak a BLE-k mellett, amelyek ugyan kizárólag valamilyen speciális funkció elvégzésére alkalmasak, de azt a funkciót nagyon hatékonyan valósítják meg. A legelső ilyen erőforrások - az FPGA áramkörök elsődleges alkalmazási területe, a digitális jelfeldolgozás igényeinek megfelelően - a **blokk RAM** és **DSP modulok** voltak.

A BLE-kből létrehozható ugyan tetszőleges méretű memória, hiszen minden BLE tartalmaz egy tároló elemet, de éppen amiatt, hogy egy BLE mindössze egyetlen bit tárolására képes, egy nagy méretű memória összeállítása megengedhetetlenül sok BLE-t igényelne, ráadásul a tárcapacitás növekedésével az egyes bitek közötti összeköttetések megvalósítása is egyre nehezebbé válik. E probléma megoldására az FPGA-kba **blokk RAM modulokat** építenek be, amelyek **nagysebességű** és a BLE-kből felépíthetőhöz képest **nagy kapacitású** (MBit nagyságrend) **memóriák**. Bár funkciójuk alapvetően kötött, a rugalmas felhasználhatóság érdekében **számtalan tulajdonságuk konfigurálható**. Általában single- vagy dual-port működésre képesek, kimeneteik szinkron működésűek, az általuk tárolt vektorok mérete és száma gyakorlatilag tetszőlegesen megválasztható (természetesen vannak eszköztől eszközre változó felső korlátok).

Ugyancsak nagyszámú BLE-re lenne szükség a DSP alkalmazásokban igen gyakran előforduló **szorzás műveletet** megvalósító áramkörök megvalósításához, ezért az FPGA-k viszonylag nagy számban tartalmaznak ún. **DSP slice-okat**, vagyis olyan BLE-eket, amelyek **blokk szorzó**, **szorzó-akkumulátor**, **barrel-shifter** és ezekhez hasonló áramköri részleteket tartalmaznak.

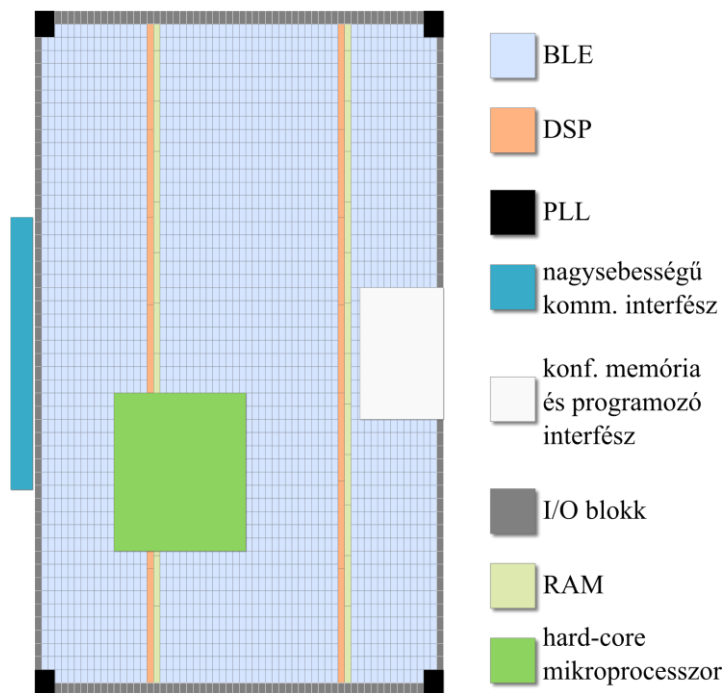
A nagy kapacitású FPGA-k általában egy vagy több, **hard-makróként beépített mikroprocesszor magot** is tartalmaznak. Ezek a mikroprocesszorok általában nagy számítási teljesítményű RISC magok, amelyek rendszerint adminisztratív és vezérlési feladatokat látnak el.

Az FPGA erőforrásokból megvalósított alkalmazás-specifikus adatfeldolgozó rendszerek és hard-makróként beépített mikroprocesszorok ún. **SoPC** (*System on a Programmable Chip*) rendszert alkotnak, amelyeknek a feladattól függően adott esetben a külvilággal is nagy sávszélességű kapcsolatot kell megvalósítaniuk. E célból az FPGA-k a **nagysebességű kommunikációs interfészek** (*high-speed transceiver*) bizonyos áramköreit is tartalmazzák, amelyek nemcsak összetettségük miatt nem lennének megvalósíthatók BLE-kből, hanem a nagy átviteli sebességből adódó magas órajelfrekvenciák miatt bizonyos részleteik analóg megoldásokat igényelnek.

Az FPGA-k kizárólag digitális rendszerek megvalósítására képesek, amelyekben kiemelt szerepe van az egyes regiszterek szinkronitását biztosító **órajelnek** és a **reset** jeleknek. Speciális célú erőforrásnak tekinthetők az e két globális jel előállítását végző **PLL-ek** (*Phase Locked Loop*) és az azok terjesztéséért felelős **órajel- és reset hálózat**, amelyek külön erre a célra szolgáló,

impedanciaillesztett és kiegyensúlyozott (az órajel-elcsúszást, az ún. **clock skew-t minimalizáló**) vezetékvezési hálózatok.

A 3-9. ábra egy mai modern FPGA sematikus ábráját mutatja, feltüntetve a speciális célú erőforrásokat (az áttekinthetőség céljából az ábra nem tartalmazza a huzalozási erőforrásokat).



3-9. ábra Erőforrástípusok egy mai modern FPGA-ban

3.2. Az FPGA-alapú áramkörösintézis folyamata

A szintézis FPGA technológia esetén ugyanazokat a részfolyamatokat foglalja magában, mint standard cellás ASIC esetén, de a generikus kapuszintű leírás elemeinek egy technológiai könyvtár elemeire való leképezése, illetve a fizikai tervezés lépései az eltérő célarchitektúra miatt más jelentenek.

- **Mapping:** A generikus kapuszintű modell egyes elemeinek (kapuk, flip-flopok) az alkalmazott FPGA eszköz erőforrásait (LUT, FF) feleltetjük meg.
- **Csoportképzés (packing):** Az FPGA erőforrások csoportokat alkotnak. Az egyazon csoportba tartozó erőforrások egymáshoz közel helyezkednek el és kis késleltetésű, rövid vezetékkel egymáshoz kapcsolhatók. Ha két összekapcsolandó logikai kapu más-más csoportban van, akkor a köztük lévő vezeték késleltetése lényegesen nagyobb. Hierarchikus huzalozási architektúrát alkalmazó FPGA-k esetén különösen nagy szerepe van a *packing* algoritmusnak, amely igyekszik minimalizálni a csoportok közötti vezetékvezés mennyiségét.
- **Elhelyezés (placement):** Az elhelyezés során a már technológiafüggő, LUT-okból és flip-flopokból álló kapuszintű modell egyes elemeihez a kiválasztott eszköz egy-egy konkrét erőforrását rendeljük.

- **Huzalozás (routing):** A huzalozás során meghatározzuk a kapcsolómátrixok megfelelő konfigurációját, amelyek így a kívánt összeköttetéseket létrehozzák a LUT-ok és flip-flopok között.
- **Konfigurációs memória tartalmának előállítása (bitstream generation):** A kapcsolómátrixok és egyéb konfigurálható erőforrások beállításait tartalmazó adathalmaz egy soros interfészen (általában JTAG) keresztül kerül az eszközbe. A szintézis utolsó lépésében tehát előállítjuk az eszközre jellemző bitfolyamot.

3.3. FPGA-specifikus tervezési megfontolások

Az FPGA áramkörök konfigurálhatóságukból adódóan **nagy rugalmasságot biztosítanak**, de speciális felépítésüknek köszönhetően az általuk **megvalósított áramkörök bizonyos tulajdonságai kedvezőtlenek egy standard cellás ASIC megoldáshoz képest**. A logikai alapelemek közötti összekötteteshálózat hatalmas terhelő kapacitást képvisel. A nagy méretű - ez által nagy kapacitású - vezetékek potenciálja - meghatározott töltő- és kisütőáramok mellett - csak jelentős idő alatt változtatható meg, ami nagy késleltetésű összeköttetéseket jelent. Mindezek mellett az absztrakt kapusintű modell egy-egy összeköttetése fizikailag akár több vezetékseggel is tartalmazhat, amelyek között az összeköttetést a 3-5. ábrán láthatóhoz hasonló struktúrák valósítják meg. Az ilyen multiplexer-alapú kapcsolómátrixokban a vezetékseggemensek határán lévő transmission-gate-ek tovább növelik a késleltetést. Mindezek a jellegzetességek azt eredményezik, hogy **az FPGA-val megvalósított áramkörök lassabb működésre képesek, mint az azonos funkciót ellátó, standard cellás technológián megvalósított megfelelőik. Fogyasztás tekintetében** - ugyancsak az óriási mennyiségű vezetékvezési erőforrás miatt - az FPGA-k szintén **alulmaradnak a standard cellás áramkörökkel szemben**¹.

Mivel az FPGA áramkörök fizikai terve már rendelkezésre áll, a felhasználónak **nincs lehetősége az egyes logikai elemek közötti késleltetés explicit befolyásolására**. Ez azt jelenti, hogy egyes, standard cellás ASIC technológián szokásos tervezési módszerek **nem használhatók** FPGA technológia esetén. Ilyen tervezési fogások pl. a **késleltető láncok, aszinkron részáramkörök** és **kombinációs hurkok**. Ezek mindegyike a logikai kapuk közötti vezetékvezés késleltetésének finomhangolásán alapul. Standard cellás ASIC esetén ezeket a késleltetéseket a fizikai tervezés során kézben lehet tartani, de FPGA esetén - bár maga a szintézis eszköz rákényszeríthető e struktúrák leszintetizálására (lásd *Digitális rendszertervezés V.: ring-oszcillátor*) - a szintézis többszöri lefuttatása mindig más és más eredményt ad a vezeték-késleltetésekre vonatkozóan, ami ilyen esetekben a funkcionalitás megváltozását is jelentené.

Ugyancsak a fizikai tervezés hiánya miatt FPGA technológián általában **erőteljesebben alkalmazzák a pipeline technikát**, vagyis a mély, sokszintű logikai szerkezetek regiszter

¹ A standard cellás ASIC és FPGA-alapú áramkörök effajta összehasonlítása természetesen csak akkor jogos, ha azonos technológiát feltételezünk. Egy 14 nm-es tri-gate technológián megvalósított FPGA nagyságrendekkel jobb lehet, mint egy 350 nm-es standard cellás ASIC, akár fogyasztásról, akár sebességről van szó.

rétegekkel való felbontását kisebb késleltetésű, kevesebb logikai kaput tartalmazó rész-logikákra. Standard cellás ASIC technológia esetén nem ritka az akár 15 szintű logikai hálózat, mert a fizikai tervezés során gondoskodni lehet arról, hogy az azt alkotó kapuk közel legyenek egymáshoz. **FPGA szintézis esetén a kapuk közötti távolság nem determinisztikus.**

Érdekes különbség a standard cellás ASIC és FPGA-alapú áramkörösintézis között, hogy míg standard cellás esetben az egyazon funkciót megvalósító áramkörök maximális órajel-frekvencia és erőforrásigény szempontjából sokféleképpen megvalósíthatók, addig FPGA esetén e paraméterek tekintetében a mozgástér lényegesen szűkebb. Ennek oka, hogy egy standard cellakönyvtár egy adott logikai funkcióhoz egy egész sor cellát kínál, amelyek méretükben (meghajtóképességükben) és sebességükben nagymértékben eltérhetnek. A maximális órajel-frekvenciát meghatározó kritikus út optimalizációja során egy standard cellás szintézis eszköznek a cellakönyvtár nagy mozgásteret biztosít, míg FPGA esetén egy regiszter meghajtóképessége legfeljebb annak duplikálásával növelhető. A regiszterduplikálás azonban a regiszterszám növekedésével egyre kevésbé hatékony a beiktatandó vezetékezés miatt.

Standard cellás ASIC áramkörökben a különböző **órajeltartományok száma általában nagyobb, mint FPGA esetén.** Ennek oka, hogy az FPGA-kban az órajel terjesztését előre elkészített, erre a célra dedikált vezetékhalózat, az órajelfa végzi. Az **órajelfa elemei** (bufferek, deskew áramkörök, kis késleltetésű vezetéksegmensek), illetve a különböző **órajelek előállítását végző PLL-ek is korlátozott mennyiségben állnak rendelkezésre**, így FPGA esetén célszerű az aszinkron órajeltartományok számát minimalizálni.

A fenti felsorolás korántsem teljeskörű (további részletek: [5][6]), de jól mutatja azt, hogy **a technológia jellegzetességei már az RTL modellezés szintjén figyelembe veendők**, így az az állítás, mely szerint az RTL modell még teljesen technológiafüggetlen, csak részben bizonyul igaznak. Tetszőleges RTL modell szintetizálható ugyan standard cellás ASIC és FPGA technológiára egyaránt, azonban a szintetizált áramkör minősége és megbízhatósága nagymértékben függ attól, hogy az RTL tervező mennyire veszi figyelembe a két technológia eltéréseiből adódó különbségeket.

3.4. Ajánlott irodalom

- [1] Clive "Max" Maxfield, **FPGAs: Instant Access**, 1st edition, ELSEVIER, 2008
- [2] Gina R. Smith, **FPGAs 101**, ELSEVIER, 2010
- [3] Ian Kuon, Russel Tessier, Jonathan Rose, **FPGA Architecture: Survey and Challenges**, Foundations and Trends in Electronic Design Automation, vol. 2, no. 2, pp. 135-253, 2007
- [4] Roger Woods, John McAllister, Gaye Lightbody, Ying Yi, **FPGA-based Implementation of Signal Processing Systems**, John Wiley and Sons, Ltd., 2008
- [5] Jari Nurmi (ed.), **Processor Design - System-On-Chip Computing for ASICs and FPGAs**, Springer, 2007
- [6] Steve Kilts, **Advanced FPGA Design - Architecture, Implementation, and Optimization**, John Wiley & Sons, Inc., 2007