

Dr. Vörösházi Zsolt:

Beágyazott rendszerek fejlesztése (FPGA)



**A felsőfokú informatikai oktatás
minőségének fejlesztése,
modernizációja**

TÁMOP-4.1.2.A/1-11/1-2011-0104



Főkedvezményezett:
Pannon Egyetem
8200 Veszprém
Egyetem u. 10.

Kedvezményezett:
Szegedi Tudományegyetem
6720 Szeged
Dugonics tér 13.



Frissítve: 2019. szeptember 10.

Beágyazott rendszerek fejlesztése (FPGA)

Dr. Vörösházi Zsolt

1. Bevezetés. Beágyazott rendszerek. FPGA- k és fejlesztő környezetek.



- Bevezetés – Beágyazott rendszerek
- FPGA (**F**ield **P**rogrammable **G**ate **A**rrays)
- Használt fejlesztő hardverek és eszközök:
 - Digilent **ZyBo** fejlesztő kártya
- Használt fejlesztő szoftverek:
 - Xilinx Vivado Design Suite (**2018.3**)
 - Vivado (Embedded FW) + SDK (Software Dev. Kit)

- 1. Bevezetés – Beágyazott rendszerek, FPGA-k, Digilent ZYBO fejlesztő kártyák és eszközök**
2. Beágyazott Rendszer fejlesztő szoftverkörnyezet (Xilinx Vivado Embedded Development) áttekintése
3. Beágyazott alap tesztrendszer (BSB - Base System Builder and Board Bring-Up) összeállítása
4. Perifériák hozzáadása (IP adatbázisból) az összeállított beágyazott alaprendszerhez. Saját periféria hozzáadása az összeállított beágyazott alaprendszerhez
5. Szoftver alkalmazások fejlesztése, tesztelése, hibakeresése (debug) Xilinx Vivado SDK (Software Development Kit) használatával

Ajánlott és felhasznált irodalom



- *Fodor Attila, Dr. Vörösházi Zsolt: Beágyazott rendszerek és programozható logikai alkatrészek (TÁMOP 4.1.2) c. Egyetemi jegyzet (2011)*
-  http://www.tankonyvtar.hu/hu/tartalom/tamop425/0008_fodorvoroshazi/Fodor_Voroshazi_Beagy_0903.pdf
(**1. Beágyazott rendszerek, 2.9 Buszok, beágyazott processzorok fejezetrészek áttekintése ajánlott!!**)
- Xilinx Teaching Materials
 <https://www.xilinx.com/support/university/course-materials.html>
- Digilent **ZYBO** FPGA kártya adatlapok
 - ZYBO kártya:
 <https://store.digilentinc.com/zybo-zynq-7000-arm-fpga-soc-trainer-board/>
 - PMOD periféria modulok:
 <https://store.digilentinc.com/pmod-modules-connectors/>

- Xilinx hivatalos oldala:
 -  <http://www.xilinx.com>
- EE Journal – Electronic Engineering:
 -  <http://www.eejournal.com/design/embedded>
- EE Times:
 -  <http://www.eetimes.com/design/embedded>

Bevezetés

BEÁGYAZOTT RENDSZEREK

- A beágyazott rendszer (**Embedded System**) a (számítógépes) **hardver-** és **szoftver**elemeknek kombinációja, amely kifejezetten egy adott funkciót, *specifikus* (vezérlési) feladatot képes ellátni, szemben az általános célú számítógép rendszerekkel.
- A beágyazott rendszerek olyan számítógépes eszközöket tartalmazhatnak, amelyek alkalmazás-orientált célberendezésekkel (**ASIC, ASSP, FPGA, MCU, MPU, DSP**, stb.), vagy komplex alkalmazói rendszerekkel (akár OS) szervesen egybeépülve akár azok **autonóm** működését is képesek biztosítani.
- A **programozható** beágyazott rendszerek olyan programozói interfésszel vannak ellátva, amelyek általában sajátos szoftver (firmware) fejlesztési stratégiákat és technikákat követelnek meg.

Néhány fontos alkalmazási terület



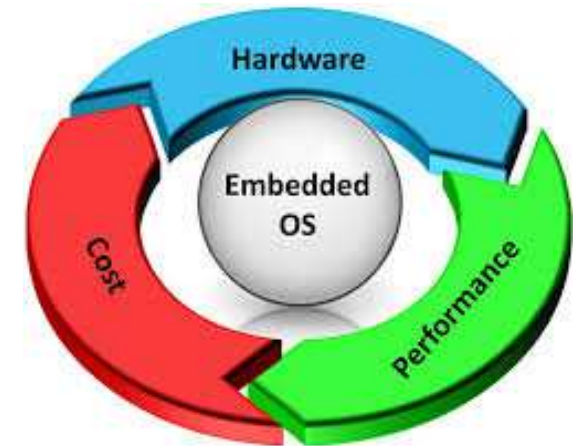
- Autóipari alkalmazások: beágyazott elektronikus vezérlők
 - Biztonságkritikus: központi elektronikai vezérlő (ECU), motorvezérlés, fékrásegítő, sebességváltó, blokkolásgátló vezérlés (ABS), kipörgés-gátló (ESP) légzsák
 - Utas központú (komfort) rendszerek: szórakoztatás, ülés/tükör ellenőrzés stb.
- Repülőgép-ipari és védelmi alkalmazások
 - Repülésirányító rendszerek (fedélzeti navigáció, GPS vevő), hajtómű vezérlés, robotpilóta
 - Védelmi rendszerek, radar rendszerek, rádió rendszerek, rakétavezérlő rendszerek
- Gyógyászati berendezések:
 - Orvosi képfeldolgozás
 - Jelmonitorozás (PET, MRI, CT)
- Hálózati/ telekommunikációs rendszerek (modem, router stb.)
- WSN: Vezeték nélküli szenzorhálózatok (motes)
- IoT: Intelligens, vagy smart rendszerek
- Háztartási gépek, ill. fogyasztói elektronika
 - mobiltelefon, PDA, PNA, digitális kamera, nyomtató stb.



Általános követelmények



- Dedikált funkció
 - Jól körülhatárolt (alkalmazás specifikus) funkció(k) támogatása
- Szigorú követelmények
 - Alacsony költség (**C**ost)
 - Gazdaságosság (**E**conomy) - lehetőleg minimális alkatrészből épüljön fel
 - Gyors működés (**S**peed)
 - Alacsony disszipáció (**P**ower)
- Valós idejű (real-time) működés és válasz
 - a környezetet folyamatos monitorozása, és beavatkozás
- Hardver-, és szoftver részek elkülönült, de együttes tervezése (co-design), tesztelése (co-simulation), ellenőrzése (co-verification)



- **Idő:** Egy bekövetkező esemény kezelését a beágyazott rendszer egy *meghatározott* időn belül kezdje el.
- **Biztonság:** olyan rendszer vezérlése, amely hibás működés esetén egészségkárosodás, és komoly anyagi kár nélkül kezeli a bekövetkező eseményt.

E filozófia mentén a beágyazott rendszerek két *alcsoportját* lehet definiálni:

- **Valós idejű rendszer (v. idő kritikus):** melynél az időkövetelmények betartása a legfontosabb szempont,
- **Biztonságkritikus rendszer:** melynél a biztonsági funkciók sokkal fontosabbak, mint az időkövetelmények betartása.

Megjegyzés: A valóságban nem lehet ilyen könnyedén a beágyazott rendszereket csoportosítani, mert lehetnek olyan valós idejű rendszerek is, melyek rendelkeznek a biztonságkritikus rendszerek bizonyos tulajdonságaival. Szabványok és a törvények szabályozzák azt, hogy milyen alkalmazásoknál kell kötelezően biztonságkritikus rendszert alkalmazni (pl. ADAS ISO 26262).

A követelmények szigorúsága alapján kétféle valós-idejű (real-time) rendszert különböztethetünk meg:

- **„hard” real-time rendszer:** *szigorú* követelmények vannak előírva, és a *kritikus* folyamatok meghatározott *időn* belül kell, hogy feldolgozásra kerüljenek,
- **„soft” real-time rendszer:** a követelmények kevésbé szigorúak, és a *kritikus* folyamatokat a rendszer mindössze nagyobb *prioritással* dolgozza fel.

A (valós-idejű) operációs rendszerek (**OS/RTOS**) számára is kritikus feladat az ütemezés és az *erőforrásokkal való optimális gazdálkodás*. Mivel minden rendszer, valamilyen periféria segítségével kommunikál a környezetével, ezért fontos a perifériák valós idejű rendszer követelményeinek megfelelő módon történő kezelése: a válaszidő betartásához az eseményt lekezelő *utasítás sorozatot* végre kell hajtani. Az utasítássorozat lefutása *erőforrásokat* igényel, melyeket az operációs rendszernek kell biztosítani, hogy hozzá tudja rendelni az időkritikus *folyamatokhoz*.

A processzorok **ütemezésének** következő szintjeit lehet megkülönböztetni:

- Hosszú-távú (long term) ütemezés vagy munka ütemezés,
- Közép-távú (medium term) ütemezés,
- Rövid-távú (short term) ütemezés.

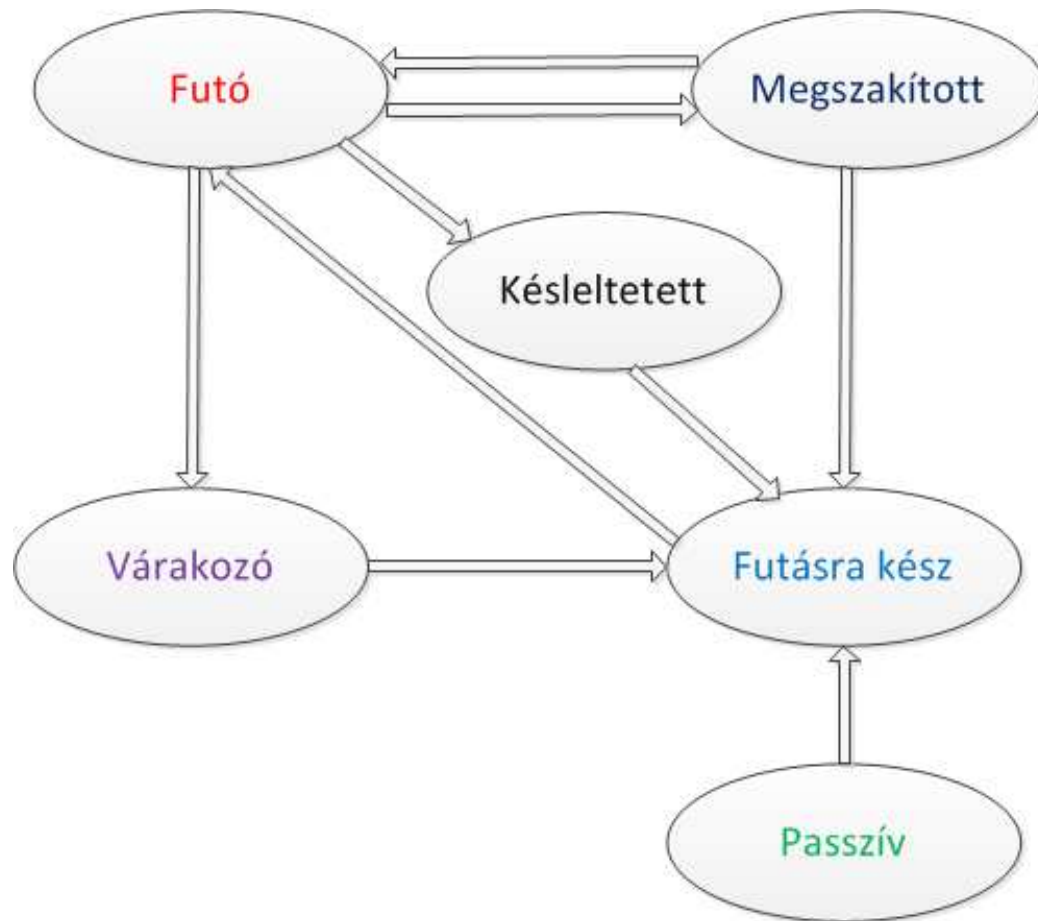
Az operációs rendszerek magja (kernel) tartalmazza az ütemezőt.

- **A hosszú-távú ütemezés** feladata, hogy a *háttértáron* várakozó, még el nem kezdett munkák közül meghatározza, melyek kezdjenek el futni, a munka befejeződésekor ki kell választania egy új elindítandó munkát. A hosszú-távú ütemezést végző algoritmusnak ezért *ritkán* kell futnia.
- **A közép-távú ütemezés** az időszakos *terhelésingadozások*at hívatott megszüntetni, hogy a nagyobb terhelések esetében ne legyenek időtúllépések. A középtávú ütemező algoritmus ezt úgy oldja meg, hogy bizonyos (nem időkritikus) folyamatokat *felfüggeszt*, majd *újraaktivál* a rendszer terhelésének a függvényében. Folyamat felfüggesztése esetén a folyamat a *háttértáron* tárolódik, az operációs rendszer elveszi a folyamattól az erőforrásokat, melyeket csak a folyamat újraaktiválásakor ad vissza a felfüggesztett folyamatnak.
- **A rövid-távú ütemezés** feladata, hogy kiválassza, hogy melyik futásra kész folyamat *kapja meg a processzort*. A rövidtávú ütemezést végző algoritmus *gyakran és gyorsan* fut le, ezért az operációs rendszer mindig a *memóriában* tartja az ütemező kódját.

Az ütemezéssel és a programokkal kapcsolatban a következő alapfogalmak értelmezhetők:

- **Task:** Önálló részfeladat.
- **Job:** A task-ok kisebb, rendszeresen végzett részfeladatai.
- **Process:** A legkisebb futtatható programegység, egy önálló ütemezési entitás, amelyet az OS önálló programként kezel. Van saját (védett) memória területe, mely más folyamatok számára elérhetetlen. A task-okat folyamatokkal implementálhatjuk.
- **Thread:** Saját memóriaterület nélküli ütemezési entitás, az azonos szülőfolyamathoz tartozó *szálak* azonos memóriaterületen dolgoznak.
- **Kernel:** Az operációs rendszer alapvető eleme, amely a task-ok kezelését, ütemezést és a task-ok közti kommunikációt biztosítja. A kernel kódja hardver-függő (device driver), valamint hardverfüggetlen rétegekből együttesen épül fel.

Task állapotok változása



- **Passzív (Dormant):** Passzív (nyugvó) állapot, amely jelentheti az inicializálás előtti vagy felfüggesztett állapotot.
- **Futásra kész (Ready):** A futásra kész állapotot jelöli. Fontos a task prioritási szintje és az is, hogy az éppen aktuálisan futó task milyen prioritási szinttel rendelkezik, ezek alapján dönti el az ütemező, hogy elindítja-e a taskot.
- **Futó (Running):** A task éppen tevékenyen fut.
- **Késleltetett (Delayed):** Ez az állapot akkor lép fel, mikor a task valamilyen *időintervallumig* várakozni kényszerül. Rendszerint szinkron időzítő (*timer*) szolgáltatás hívása után következik be.
- **Várakozó (Waiting):** A task egy meghatározott eseményre várakozik. (Ez rendszerint valamilyen I/O periféria művelet szokott lenni.)
- **Megszakított (Interrupted):** A task-ot megszakították, vagy a megszakítás kezelő rutin éppen megszakítja a folyamatot (IRQ, INT).

Az ütemezési algoritmusoknak két fő típusa van:

- **Kooperatív** (=nem preemptív): A működési elve és alapötlete, hogy egy adott program vagy folyamat *lemond* a processzorról, ha már befejezte a futását vagy valamilyen I/O műveletre vár. Ez az algoritmus addig működik jól és hatékonyan, amíg a szoftverek megfelelően működnek (nem kerülnek végtelen ciklusba) és lemondanak a processzorról. Ha viszont valamelyik a program/folyamat nem mond le a processzorról vagy kifagy, akkor az egész rendszer stabilitását képes lecsökkenteni. A kooperatív algoritmus ezért soha nem fordulhat elő valós-idejű beágyazott operációs rendszerek esetében.
- **Preemptív**: az operációs rendszer részét képező *ütemező algoritmus vezérli* a programok/folyamatok futását. A preemptív multitask esetén az operációs rendszer elveheti a folyamatoktól a *futás jogát* és átadhatja más folyamatoknak. A valós idejű operációs rendszerek ütemezői minden esetben preemptív algoritmusok, így bármely program vagy folyamat leállása nem befolyásolja számottevően a rendszer stabilitását.

Mivel a rendszer működése közben a task-ok egymással párhuzamosan futnak ezért gondoskodni kell arról, hogy egyazon I/O perifériát, erőforrást vagy memória területet két vagy több task ne használjon egyszerre, mert abból hibás rendszerműködés alakulna ki.

A következő ismert módszerek állnak rendelkezésre:

- **Mutex (kölcsonös kizárás):** ún. „locking” mechanizmus (csak a task amelyik zárolta, oldhatja fel)
- **Szemafor (semaphore):** „signaling” mechanizmus (egyik task jelez a másiknak, hogy végzett, és átveheti az erőforrást) ~ 1 bit információ
- **Események (event flags):** melyek több bit információ kicserélésére is alkalmasak.
- **Postaláda (mailbox):** amely akár komplexebb adatstruktúra átadására is szolgálhat.
- **Sor (queue):** amely több mailbox tömbjében lévő tartalom átadására szolgál.
- **Cső (pipe vagy FIFO):** amely direkt, folyamatos (akár streaming) kommunikációt tesz lehetővé két task között.

(Beágyazott) Operációs rendszerek



Többféle csoportosítás lehetséges:

- Általános célú, vagy **beágyazott OS**
- Valós-idejű (idő-kritikus), vagy nem-időkritikus
- Nyílt forráskódú, vagy licenszelhető, stb.

Általános célú processzorok operációs rendszerei (OS):

- MS-DOS, Linux, Windows, stb.

Beágyazott processzorok *valós-idejű* operációs rendszerei (RTOS):

- Linux
- Android
- Micrium uC/OS
- QNX
- RTLinux
- **Windriver VxWorks (RT)**
- Windows Embedded, IoT, stb...



Processzorok osztályozása

- Integráltság szerint:
 - uP/CPU: hagyományos mikroprocesszorok + fizikailag különálló memória + külső I/O periféria chippek (chipset)
 - uC/MCU: mikrokontrollerek: egyetlen chipen integrálva a processzor, a memória (ált. flash), és néhány I/O periféria
 - System-on-a-Chip (SoC) : egychipes rendszer
 - Kis méret és költség, alacsony disszipált teljesítmény
- Utasítás készlet szerint:
 - RISC vs. nem RISC (CISC) ISA – utasításkészletű architektúrák
- Utasítás / Adat memória hozzáférés szerint:
 - Von Neumann (közös) vs. Harvard architektúrák (elkülönült)

Néhány architektúra típus: Intel 8051, ARM, AVR, PIC/MicroChip, MIPS, IBM PowerPC, x86 (32/64), Sun SPARC, stb.

Élenjáró *technológiák* a beágyazott rendszerek tervezéséhez és megvalósításához – processzáló egységek csoportosítása:

- **(DSP):** Digitális jelfeldolgozó processzor alapú rendszerek
- **(MCU):** Mikrovezérlő-alapú rendszerek
- **(ASIC/ASSP):** Alkalmazás specifikus (berendezés orientált) integrált áramköri technológián alapuló rendszerek
- **(FPGA):** Programozható logikai kapuáramkörök technológián alapuló rendszerek
- **(MPU/GPU):** Mikroprocesszor, vagy grafikus processzor
- **SoC = System-on-a-chip:** olyan egychipes rendszer, amely a fentieket együttesen akár integrálva is tartalmazhatja!

Fejlesztési *stratégiák*:

- HW/SW co-design: HW/SW részek együttes tervezése
- HW/SW co-verification: HW/SW részek együttes ellenőrzése és tesztelése

- Aszinkron soros kommunikációs interfészek: **RS-232**, RS-422, RS-485, stb.
- Szinkron Soros kommunikációs interfészek: **I²C**, **SPI** stb.
- Univerzális soros busz: **USB**
- Multimédia kártyák: (SD) Smart Cards, (CF) Compact Flash stb.
- Hálózat: Ethernet (1GbE / 10 GbE / 100 GbE)
- Ipari hálózati/Telepi ún. „Field-bus” protokollok: CAN, LIN, PROFIBUS, stb.
- Időzítő-ütemezők: PLL(s), Timers, Counters, Watchdog timers (WDT)
- Általános célú I/O-k (General Purpose I/O - **GPIO**): LED-ek, nyomógombok, kapcsolók, LCD kijelzők, stb.
- Analóg-Digitális/Digitális-Analóg (ADC/DAC) konverterek
- Debug portok: **JTAG**, ISP, ICSP, BDM, DP9, stb.

FPGA-alapú beágyazott rendszerek főbb tervezési lépései:

- FPGA hardver (firmware) tervezés,
 - **Beágyazható/beágyazott** processzor kiválasztása:
 - Licenzelhető Soft-core: PicoBlaze / MicroBlaze™ (Xilinx); Nios II™ (Altera), ...
 - Licenzelhető Hard-core: IBM PowerPC® (Xilinx), ARM® (Xilinx / Altera), ...
 - Nyílt forráskódú processzor magok: pl. www.opencores.org
- Programozható Perifériák kiválasztása (lásd. a jegyzet *Fejlesztő kártyák*, ill. *Beágyazott perifériák* részei),
- Eszközmeghajtók (driver) és szoftver könyvtárak (lib) generálása
 - BSP: Board Support Package,
- Alkalmazás fejlesztés:
 - Szoftver rutinok (API),
 - Megszakítás-kezelő rutinok,
 - Operációs rendszer, valós-idejű operációs rendszer.

Xilinx Vivado 2018.3

HASZNÁLT FEJLESZTŐ SZOFTVEREK

Milyen fejlesztő szoftver eszközökkel ismerkedünk meg a félév során:

- Xilinx Embedded System Design Flow Workshop / Teaching Materials segédanyagai alapján
 - (eléréséhez regisztráció szükséges - ingyenes)
- **Xilinx Vivado 2018.3**
 - **Vivado Embedded development kit (EDK)**
 - **Vivado Software Developer Kit (SDK)**
- **Digilent Adept Suite:**
 - konfigurációs bitfájl letöltéséhez, fejlesztő kártya teszteléséhez
- **Xilinx ISE Development Suite (ISE)**
 - komponensek fejlesztése magas-szintű hardver leíró nyelven (HDL)
- **Xilinx ISim (Simulation)**
 - komponensek szimulációs tesztje
- **Xilinx Embedded Development Kit (EDK)**
 - **Xilinx Platform Studio (XPS) v.14.7** használata
 - Xilinx FPGA alapú beágyazott rendszer (firmware) fejlesztő környezete
 - **Xilinx Software Developer Kit (SDK) v.14.7** használata
 - Xilinx FPGA beágyazott szoftver-rendszer fejlesztő környezete
- Xilinx ChipScope v.14.7 használata:
 - digitális logikai analizátor (Xilinx JTAG-USB)

Általános ismertetés (ismétlés)

PLD ÉS FPGA ÁRAMKÖRÖK

Miért lehet fontos a programozható logikai eszközök alkalmazása?

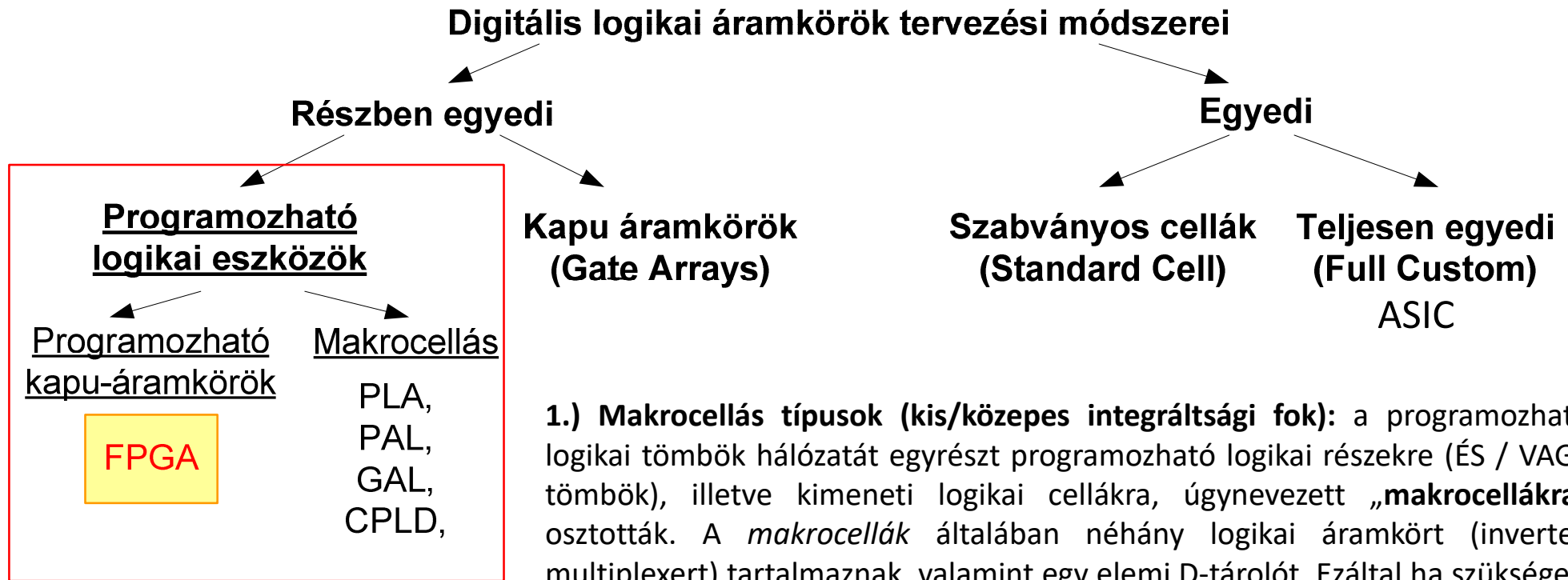


- Az 1980-as évek előtti időszakban, a digitális áramkörök logikai hálózatainak tervezése során még nem álltak rendelkezésre olyan modern fejlesztő eszközök, mint napjainkban. Nagy komplexitású (sok bemenetű - sok kimenetű) logikai kombinációs és sorrendi hálózatok tervezése éppen ezért lassú és körülményes volt, sokszor papír alapú tervezéssel, többszöri manuális ellenőrzéssel, számításokkal párosult. Fejlett szimulációs eszközökről (**CAD**) sem beszélhettünk, ezért a prototípus tervezésnél nagy volt a hibavalószínűség.
- Ma mindezek együttese automatizált módon áll rendelkezésre (**EDA** – *elektronikai tervezés automatizálása*), amely a programozható logikai architektúrák használata mellett (**PLD**), mind a nyomtatott áramkörök (**PCB**), mind pedig az alkalmazás specifikus integrált áramkörök és processzorok (**ASIC/ASSP**) relatíve gyors prototípus fejlesztését, megvalósítását (implementáció) és tesztelését (verifikáció) támogatja, valamint minimalizálja az esetlegesen előforduló hibákat. A **hardver/firmware/szoftver** részeket, együttesen és konzisztens módon lehet tervezni és tesztelni.

- Az *automatizált elektronikai tervezés (EDA)* mellett a *programozható logikai eszközök (PLD)* használata tovább csökkenti a fejlesztésre fordítandó időt, és ezáltal minimalizálja a költségeket is.
- Éppen ezért sok alkalmazási területen érdekesebb először az adott funkció kifejlesztését egy programozható logikai eszközön megvalósítani és letesztelni, majd pedig - ha teljesülnek a követelmény specifikációban megfogalmazott feltételek - következhet csak a kitesztelt funkciónak megfelelő *alkalmazás specifikus integrált áramkör (ASIC)* tervezése, gyártása, és tesztelése. Ez nagyban lerövidítheti az ASIC áramkörök fejlesztési idejét és mérsékelheti nem megtérülő költségeit (NRI).

- A Programozható logikai áramköröket (**PLD: Programmable Logic Devices**) általánosan a *kombinációs logikai* hálózatok és *sorrendi hálózatok* tervezésére használhatjuk. Azonban míg a hagyományos kombinációs logikai hálózatok dedikált összeköttetésekkel, illetve kötött funkcióval (kimeneti függvény) rendelkeznek, addig a programozható logikai eszközökben pontosan ezek változtathatók meg, az alábbi lehetséges módokon:
 - A felhasználó által *egyszer programozható* / konfigurálható logikai eszközök (OTP: One Time Programmable), amelynél a gyártás során nem definiált funkció egyszer még megváltoztatható (ilyenek pl. a korai PAL, PLA, stb. eszközök)
 - *Többször, akár tetszőleges módon* programozható logikai eszközök = *újrakonfigurálható* (ilyenek pl. a korábbi GAL, vagy a mai modern **CPLD, FPGA** eszközök)

- **Konfigurálás** – mielőtt az eszközt használni szeretnénk egy speciális (manapság általában JTAG szabványú) programozó segítségével „fel kell programozni”, azaz le kell tölteni a konfigurációs állományt (bitfájlt, vagy object fájlt). A *programozás* a legtöbb PLD esetében a belső programozható összeköttetés-hálózat fizikai típusától függően azok beállításával történik. A programozható összeköttetésekben a következő lehetséges alkatrészek, mint kapcsoló elemek találhatók:
 - *Biztosíték (Fuse)*: átégetésük után nem visszafordítható a programozási folyamat (OTP). Korábban a PAL eszközök népszerű kapcsoló elemeként használták,
 - *Antifuse technológia*: (OTP), az antifuse-os kristályszerkezetű kapcsoló elem 'átolvasztása' után egy nagyon stabilan működő összeköttetést kapunk, amely sajnos egy szintén nem visszafordítható folyamatot jelent. A technológia drága az előállításához szükséges maszk-rétegek nagy száma miatt, viszont nagyon jó a zavarvédelem (pl. úrkutatásban használt eszközök),
 - **SRAM cella + tranzisztor**: tetszőlegesen programozható (FPGA-k esetén legelterjedtebb kapcsolás-technológia), az SRAM-ban tárolt inicializáló értéktől függően vezérelhető a tranzisztor gate-elektrodája,
 - *SRAM cella + multiplexer*: tetszőlegesen programozható, az SRAM cellában tárolt értéktől függően (kiválasztó jel) vezérelhető a multiplexer,
 - *Lebegő kapus tranzisztor (Floating Gate) technológia*: elektromosan tetszőlegesen programozható, a mai **EEPROM/Flash** technológia alapja.



1.) Makrocellás típusok (kis/közepes integráltsági fok): a programozható logikai tömbök hálózatát egyrészt programozható logikai részekre (ÉS / VAGY tömbök), illetve kimeneti logikai cellákra, úgynevezett „makrocellákra” osztották. A *makrocellák* általában néhány logikai áramkört (inverter, multiplexert) tartalmaznak, valamint egy elemi D-tárolót. Ezáltal ha szükséges, regisztrálni lehet a kimeneti függvényt, majd pedig vissza lehet csatolni annak értékét a bemenetre.

- a.) PLA: Programozható logikai tömb, OTP - általában biztosítékot használ
- b.) PAL: Programozható ÉS/tömb logika, OTP – általában biztosítékot használ
- c.) GAL: Generikus tömb logika: PAL továbbfejlesztése, de már többször törölhető és programozható (nagy integráltsági fok):
- d.) CPLD: Komplex programozható logikai eszköz

2.) Programozható kapu-áramkörök (nagyon-nagy integráltsági fok):

FPGA: Felhasználó által tetszőlegesen programozható (újrakonfigurálható) kapu áramkör

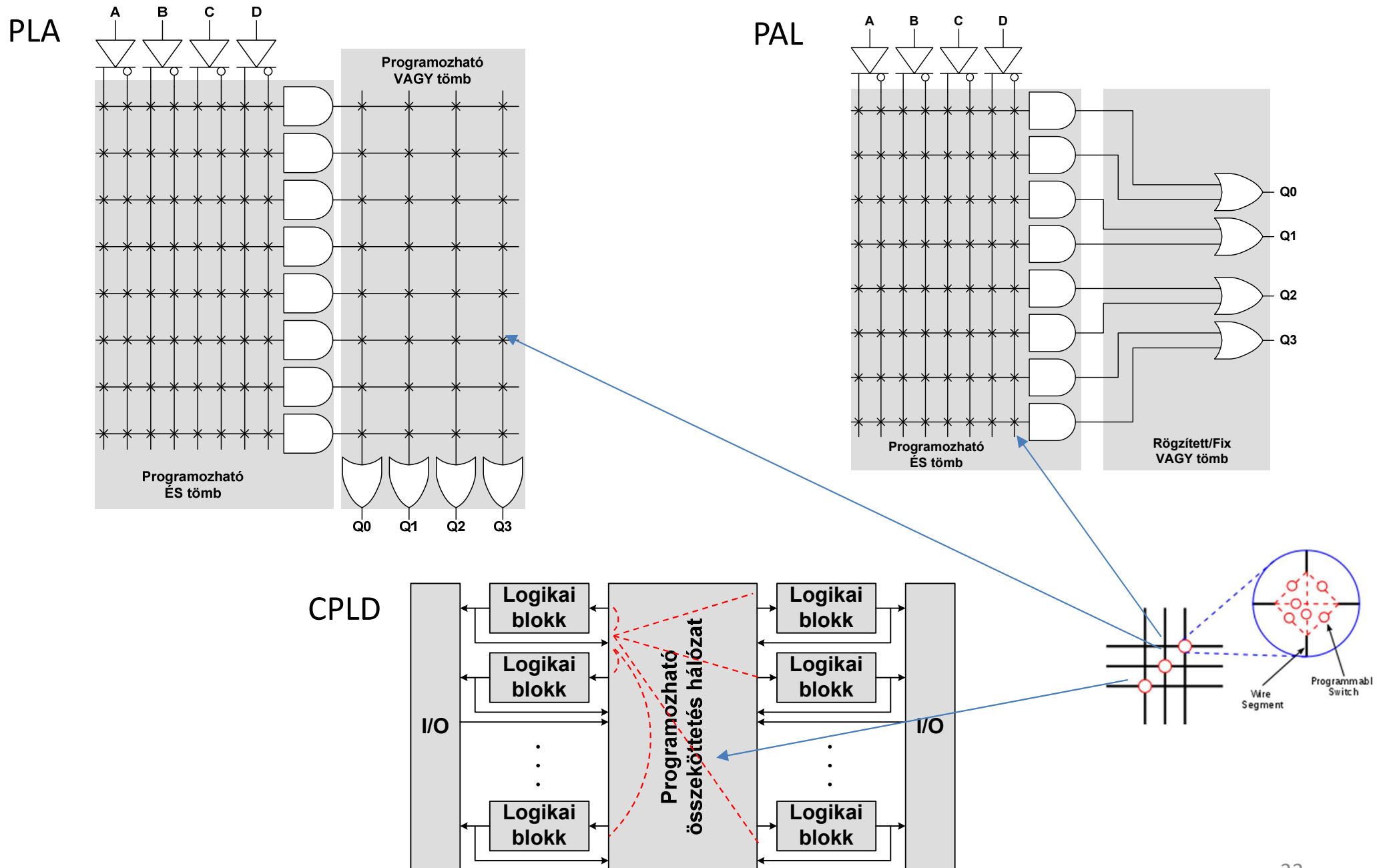
PLD-k két fő típusa



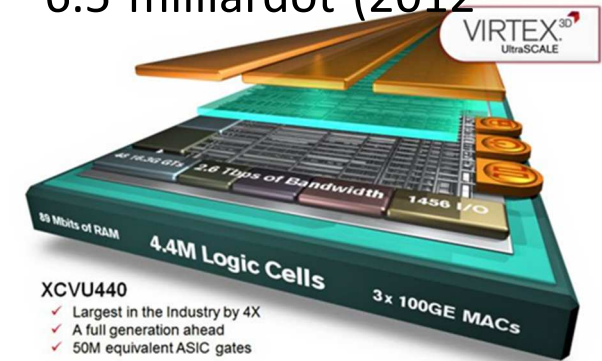
- 1.) Makrocellás PLD-k (Programmable Logic Devices):
 - PLA
 - PAL
 - GAL
 - CPLD
- 2.) FPGA (Field Programmable Gate Array):
Programozható Gate Array áramkörök
 - **XILINX** (Spartan, Virtex, Kintex, Artix) ~ 53 % !
 - **Altera/Intel-FPGA** (Stratix, Arria, Cyclone), ~ 36 %
 - MicroSemi/Actel, ~7%
 - Lattice, 3%
 - QuickLogic, < 1%
 - További kisebb gyártók termékei <1%



Makrocellás PLD-k



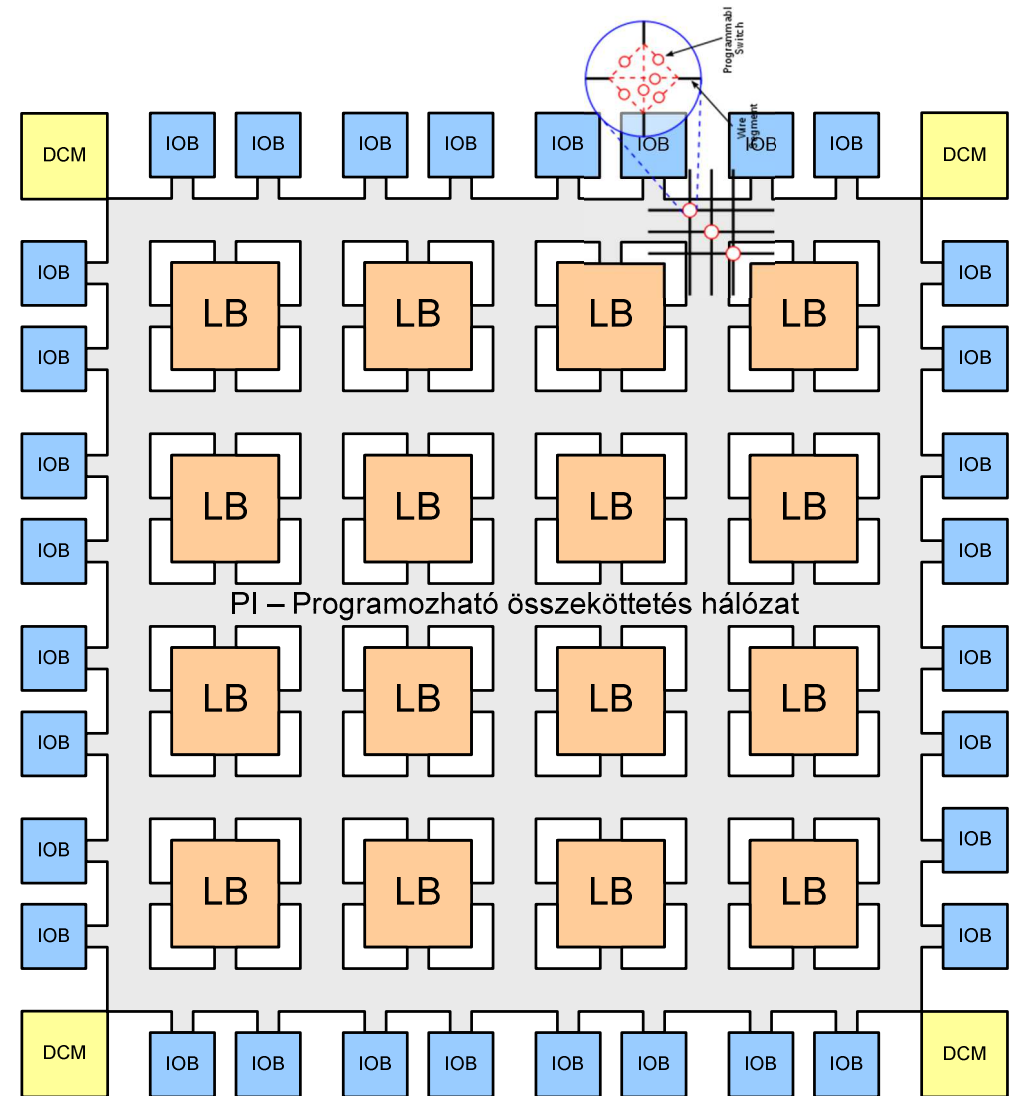
- **Field Programmable Gate Array** = „Felhasználó által tetszőlegesen/többször” programozható kapuáramkörök
 - architektúráisan tükrözik mind a PAL, ill. CPLD felépítését, komplexitásban pedig a CPLD-ket is felülmúlják. Nagy, illetve nagyon-nagy integráltsági fokkal rendelkeznek: ~10.000 - ~100.000.000 !! *ekvivalens logikai kaput* is tartalmazhat gyártótól, és sorozattól függően.
 - *Ekvivalens tranzisztorszám*
 - Xilinx Virtex-7 2000T FPGA esetén már meghaladta a ~6.5 milliárdot (2012 – 28nm), amely ~2 millió logikai cellát jelentett.
 - a kapható legnagyobb Xilinx Virtex-Ultrascale+ XCVU440 (2015 – 20nm->16nm) FPGA: ~**20 milliárd tr.** - ~**4.4** millió logikai cella! (~ 50 millió logikai kapu ekv.)
 - Intel/Altera Stratix-10 **30 milliárd tranz.** (2016, ~5.5 millió logikai cella, 4mag ARM-Cortex A53, 14nm)
 - Xilinx Virtex VU19P, **35 milliárd tranz.** (2016, ~9 millió logikai cella)



FPGA „általános” erőforrások



- **LB/CLB:** Konfigurálható Logikai Blokkok, amelyekben LUT-ok (Look-up-table) segítségével realizálhatók például tetszőleges, több bemenetű (ált. 4 vagy 6), egy-kimenetű logikai függvények. Ezek a kimeneti értékek szükség esetén egy-egy D flip-flopban tárolhatók el; továbbá multiplexereket, egyszerű logikai kapukat, és összeköttetéseket is tartalmaznak.
- **IOB:** I/O Blokkok, amelyek a belső programozható logika és a külvilág között teremtenek kapcsolatot. Programozható I/O blokkok kb. 30 ipari szabványt támogatnak (pl. LVDS, LVCMOS, LVTTTL, SSTL stb.).
- **PI:** az FPGA belső komponensei között a programozható összeköttetés hálózat teremt kapcsolatot (lokális, globális és regionális útvonalak segítségével, melyeket konfigurálható kapcsolók állítanak be)
- **DCM:** Digitális órajel menedzselő áramkör, amely képes a külső bejövő órajelből tetszőleges fázisú és frekvenciájú belső órajel(ek) előállítására



Dedikált erőforrások a következők (amelyek az FPGA típusoktól és komplexitásuktól függően nagy mértékben változhatnak):

- **BRAM:** egy-/két-portos Blokk-RAM memóriák, melyek nagy mennyiségű ($\sim \times 100\text{Kbyte}$ – akár $\sim \times 10\text{Mbyte}$) adat/utasítás tárolását teszik lehetővé, egyenként 18K / 36 Kbites kapacitással *
- **MULT / vagy DSP Blokkok:** beágyazott szorzó áramköröket jelentenek, amelyek segítségével hagyományos szorzási műveletet, vagy a DSP blokk esetén akár bonyolultabb DSP MAC (szorzás-akkumulálás), valamint aritmetikai (kivonás) és logikai műveleteket is végrehajthatunk nagy sebességgel.
- **Beágyazott processzor(ok):** **
 - Tetszés szerint konfigurálható / beágyazható ún. *szoft-processzor mag(ok)*
 - Példa: Xilinx PicoBlaze, **Xilinx MicroBlaze**, Altera Nios II stb.
 - Fixen beágyazott, ún. *hard-processzor mag(ok)*
 - Példa: IBM PowerPC 405/450 (Xilinx Virtex 2 Pro, Virtex-4 FXT, Virtex-5 FXT), **ARM Cortex-A9** (Xilinx Zynq, illetve Altera Cyclone V SoC, Arria V SoC, vagy a MicroSemi Smartfusion-1,-2 FPGA chipjei) stb.

* FPGA függő adatok (Xilinx)

** 2015-ös adatok szerint

FPGA-k létjogosultsága?



A mai modern FPGA-k a

- nagyfokú flexibilitásukkal,
- nagy számítási teljesítményükkel,
- és relatíve gyors prototípus-fejlesztési – ezáltal
- olcsó kihozatali (piacra kerülési) költségükkel

igen jó alternatívát teremtenek a mikrovezérlős (uC), illetve DSP-alapú implementációk helyett (pl. jelfeldolgozás, hálózati titkosítás, beágyazott rendszerek, stb. alkalmazásai területén). ASIC lassú fejlesztési idejét az FPGA alapú prototípus tervezéssel gyorsíthatják.

Fejlődésüket jól tükrözi a mikroprocesszorok és az FPGA áramkörüi technológia fejlődési üteme között fennálló nagyfokú hasonlóság a méretcsökkenésnek (scaling-down) - *Gordon Moore-törvénynek* megfelelően.

Manapság már több tranzisztort tartalmaznak, mint modern MPU-GPU vetélytársaik.

Xilinx Zynq-7010

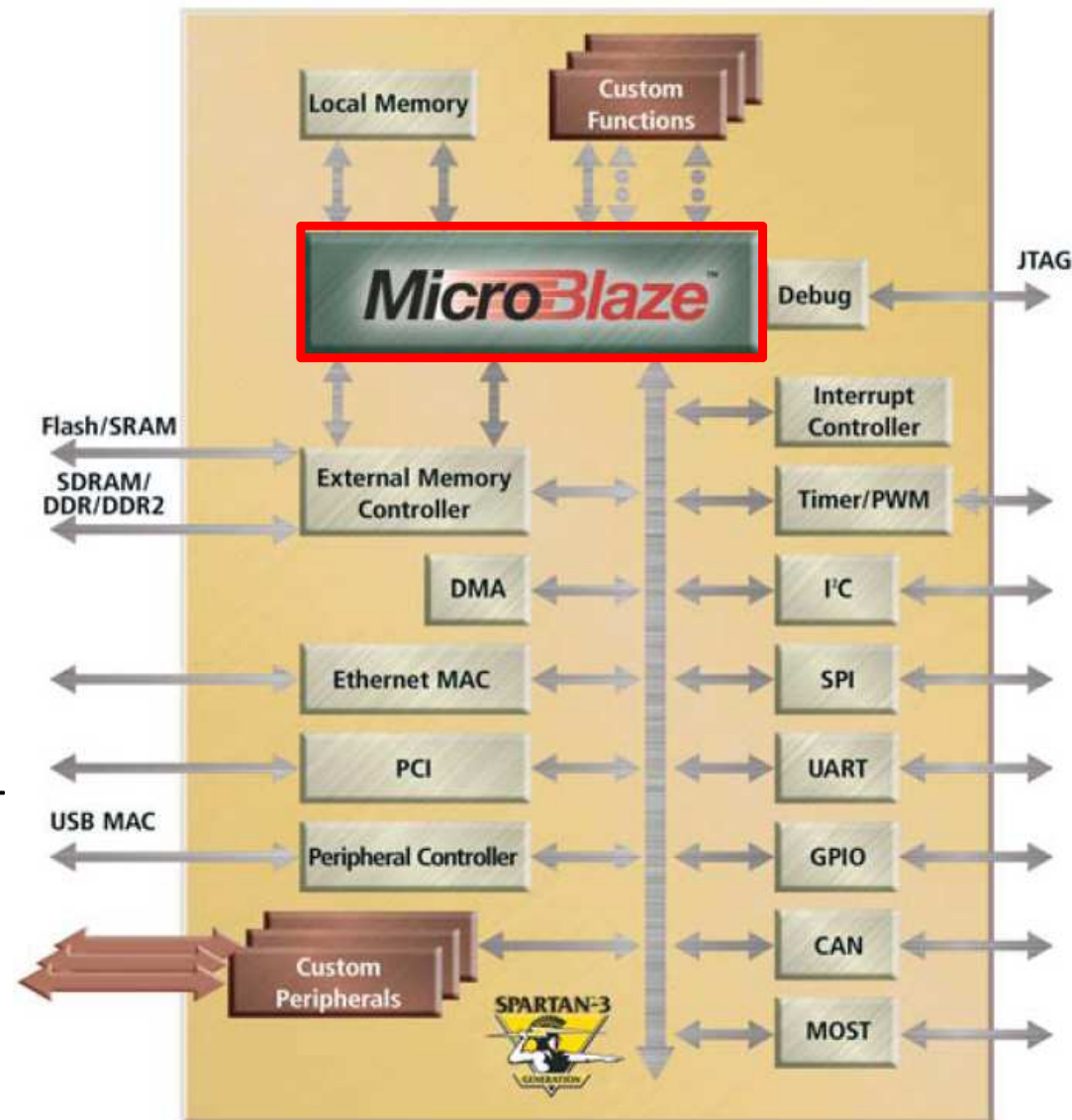
XILINX FPGA-K BEÁGYAZOTT PROCESSZORAI

- „*Beágyazható*” szoft-processzor magok:
 - Xilinx *PicoBlaze*: 8-bites (VHDL, Verilog HDL forrás)
 - *Xilinx MicroBlaze*: 32-bites (Vivado / XPS = EDK+SDK támogatás!)*
 - PLB, OPB (régi), AXI buszrendszerekhez is csatlakoztatható
 - 3rd Party: nem-Xilinx gyártók processzorai (HDL)
- „*Beágyazott*” hard-processzor magok:
 - IBM *PowerPC* 405/450 processzor (dedikált): 32-bites (EDK/SDK támogatás), PLB buszrendszerhez integrálható
 - de kizárólag Virtex II Pro, Virtex-4 FX, Virtex-5 FXT FPGA-kon!
 - *ARM Cortex-A sorozatú* processzor (dedikált): ARM AMBA-AXI buszrendszerhez integrálható
 - Xilinx Zynq **APSoC**-n integrált ARM Cortex A9 magok (32-bites)
 - Altera Stratix-10 integrált ARM Cortex A53, A72, R5 magok (64-bites)

MicroBlaze szoft-processzor mag

„Beágyazható” processzor mag*

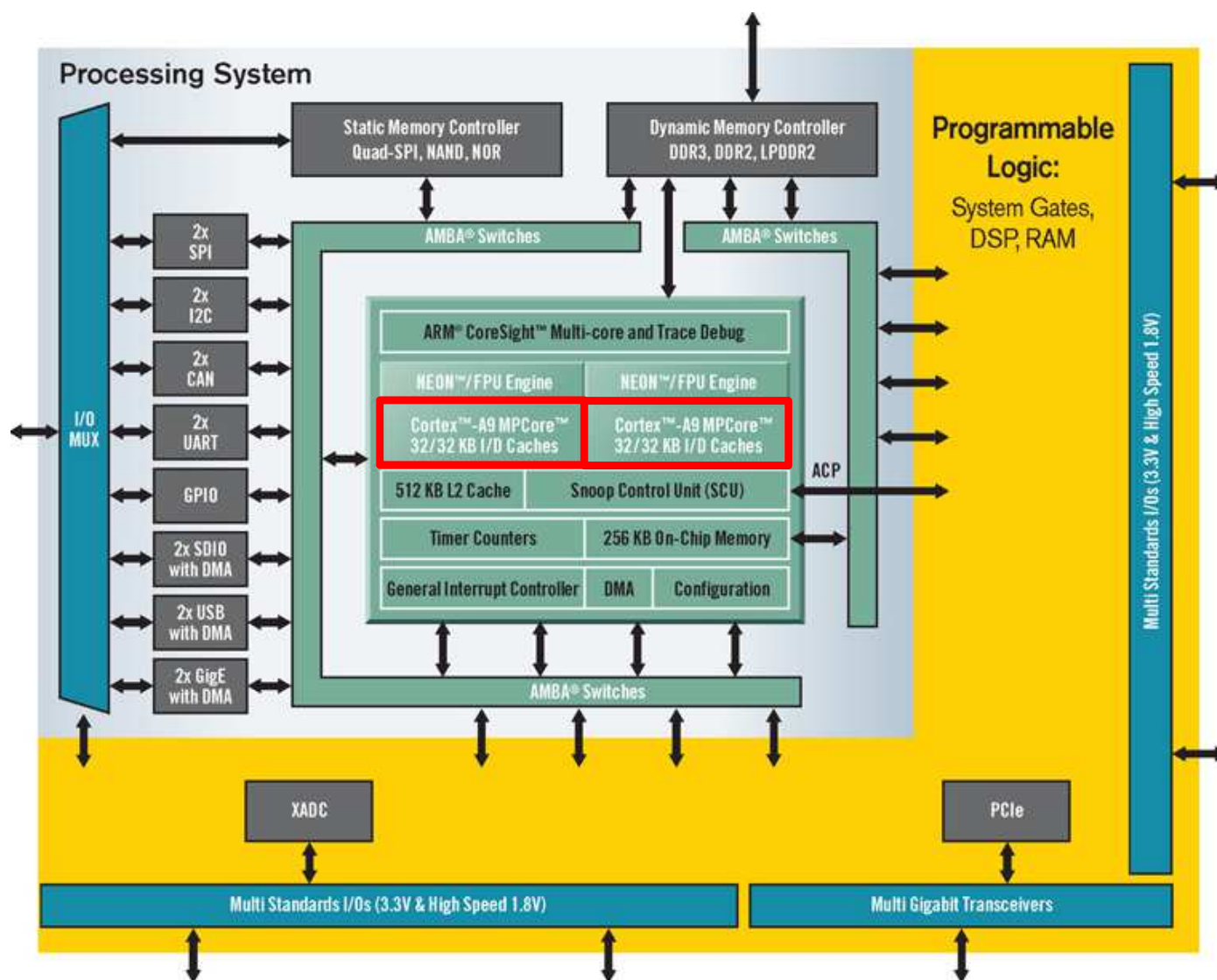
- RISC utasítás készlet architektúra
- 32-bites szoft-processzor mag
- 133+ MHz órajel (PLB/AXI** busz)
- Harvard blokk-architektúra
- Kis fogyasztás: ~ mW/MHz
- 3/5 lépcsős adatvonal pipe-line
- 32 darab 32-bites általános célú regiszter
- utasítás Cache / adat Cache
- Időzítési lehetőségek (timer)
- Sokféle periféria, kommunikációs interfész csatlakoztatható (IP core-ok)
- **Minden** Xilinx FPGA-n implementálható, melynek elegendő programozható logikai erőforrása van, és a fejlesztő szoftver támogatja!



Xilinx Zynq APSoC – ARM hard-processzor



ARM



- **Dupla ARM Cortex™-A9 beágyazott proc. 650MHz - ZYBO)**
- Neon: 32-/ 64-bit FPU
- 32kB utasítás & 32kB adat L1 Cache
- Közös 512kB L2 Cache
- 256kB on-chip memória
- Integrált DDR3, DDR2 and LPDDR2 DDR vezérlő
- Integrált 2x QSPI, NAND Flash and NOR Flash memória vezérlő
- Perifériák: 2x USB2.0 (OTG), 2x GbE, 2x CAN2.0B 2x SD/SDIO, 2x UART, 2x SPI, 2x I2C, 4x 32b GPIO, PCI Express® Gen2 x8
- Két 12-bit 1Msps ADC
- **28nm** Programozható FPGA Logika:
- 28k - 350k Logikai cella (~ 430k to 5.2M ekvivalens kapu)
- 240KB - 2180KB Block RAM
- 80 - 900 18x25 DSP szorzó (58 - 1080 GMACS -DSP teljesítmény)

APSoC = All Programmable System-On-a-Chip



ZYBO

Xilinx ZYNQ-7010 APSoC FPGA-k bemutatása



Felhasznált irodalom



- Xilinx hivatalos weboldal (2019)



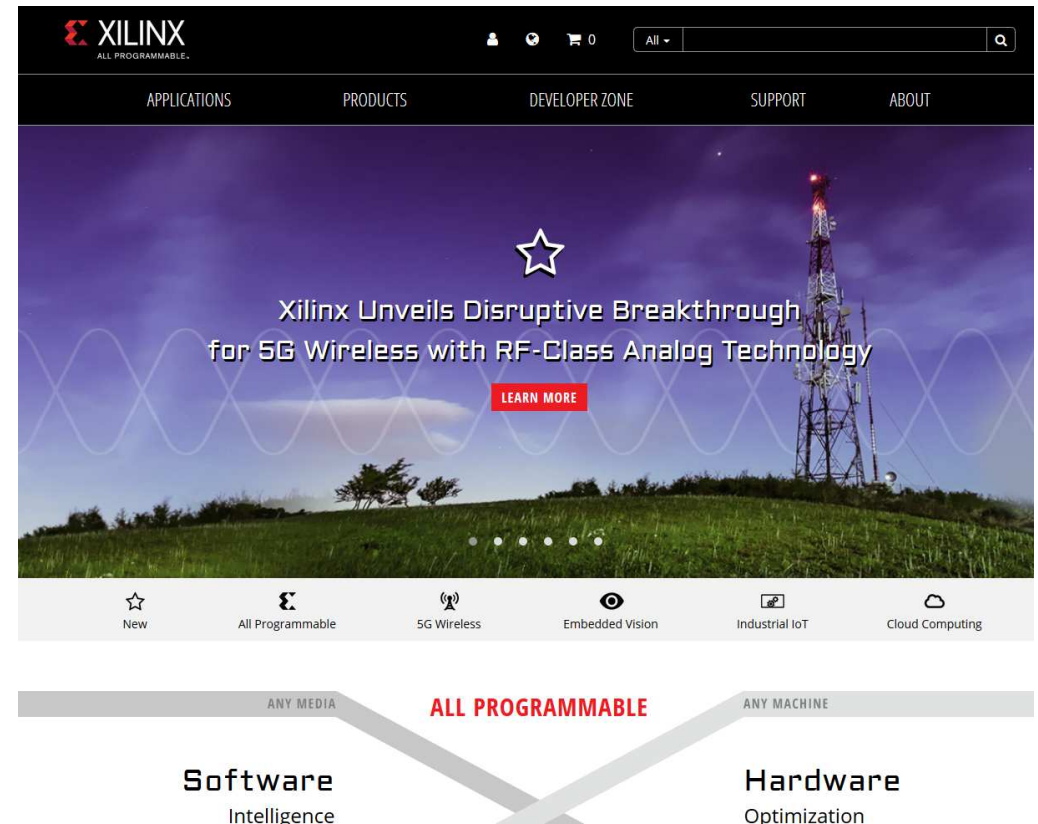
www.xilinx.com



- Xilinx Zynq adatlapok



[ds190.pdf](#) – Zynq-7000-Overview

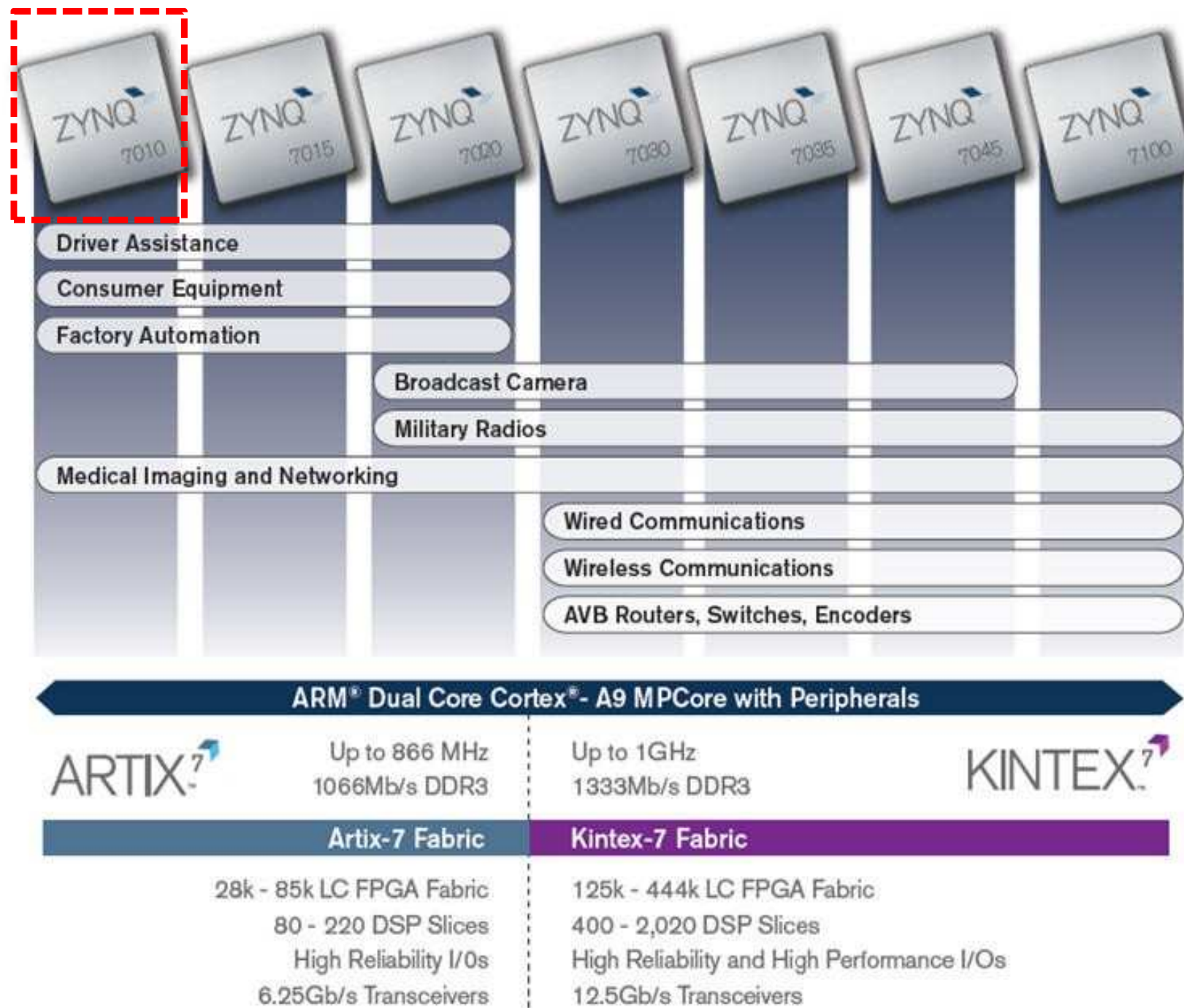


- Zynq TRM leírás (UG-585) – 1800 oldal!



https://www.xilinx.com/support/documentation/user_guides/ug585-Zynq-7000-TRM.pdf

Zynq-7000 sorozat

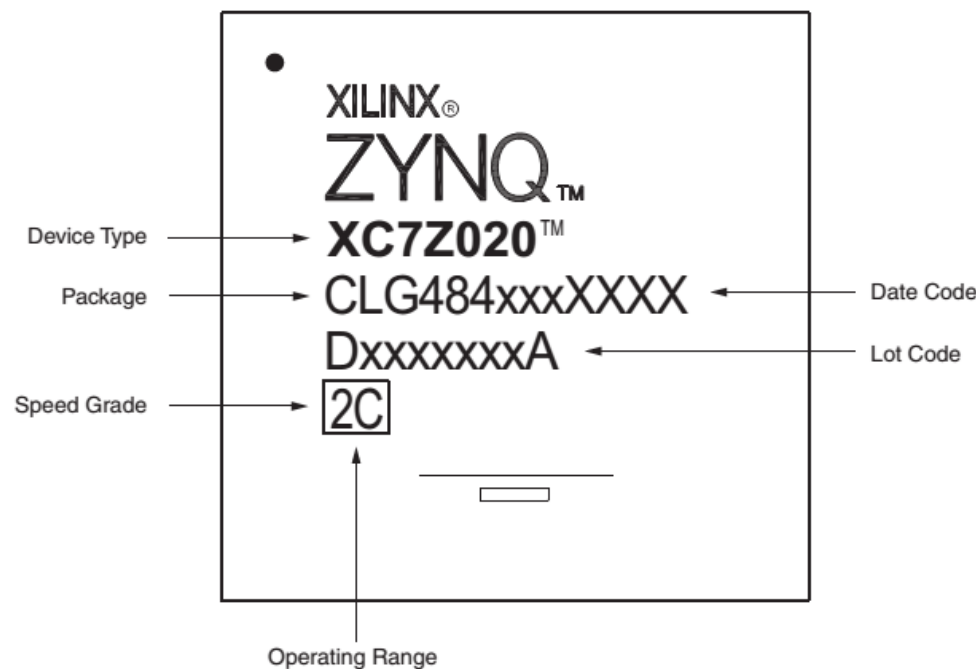


FPGA azonosítása katalógusban



- Eszköz típus (Device Type), tokozás (Package), lábszám (Pin Count) alapján együttesen
 - **ZYNQ XC7Z010-1CLG400C*** = Xilinx Zynq-7010 (~ 28.000 logikai cella ekvivalens FPGA)
 - Chip Scale (CLG400C) tokozás
 - Lábszám: 400
 - Sebesség (Speed grade =1): 1C (C - kereskedelmi célra: 0-85 C°)

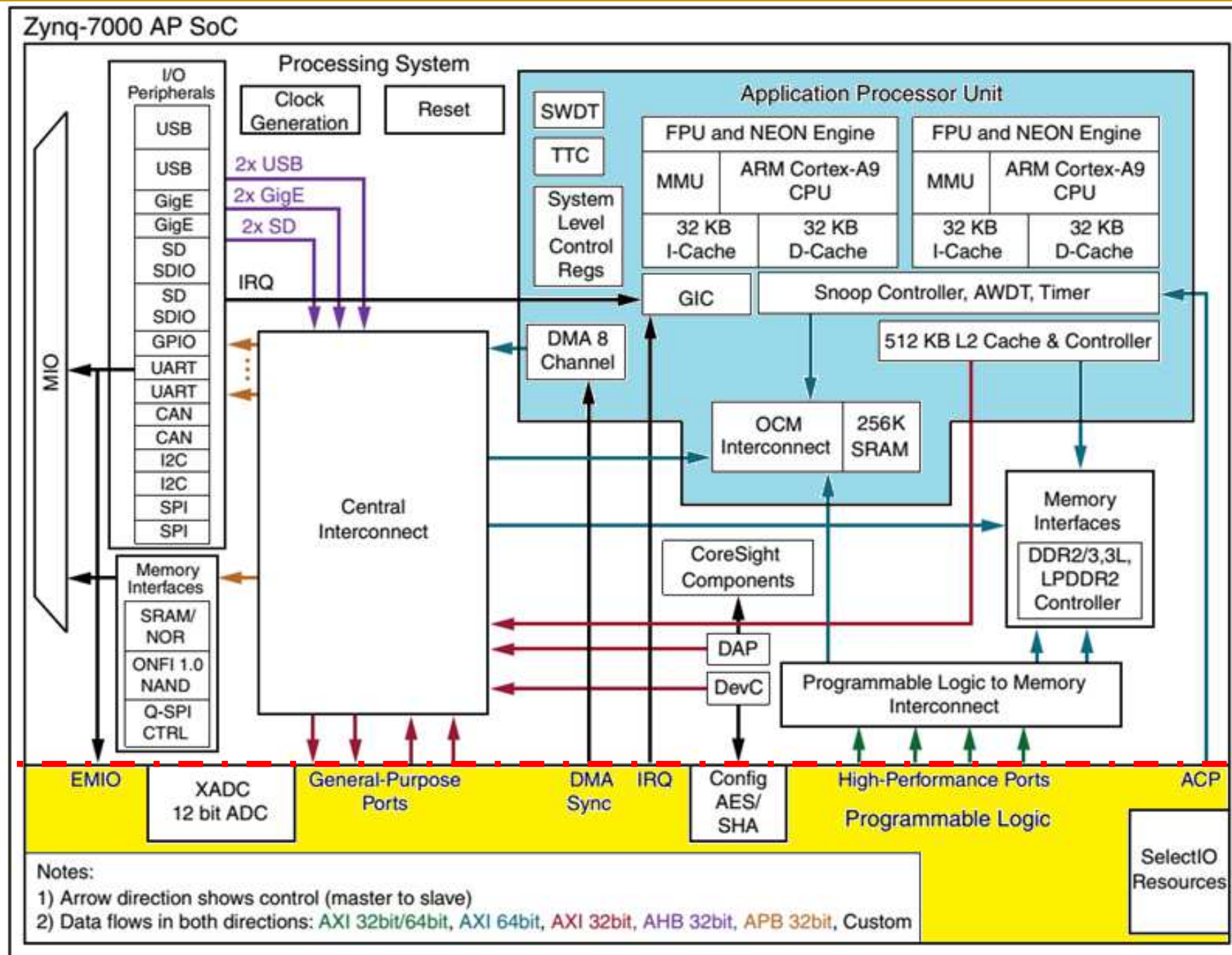
* Az FPGA paraméterek az ZYBO kártyára érvényesek



Xilinx Zynq APSoC

PS

PL



Processzor Rendszer (PS)

	Device Name	Z-7010	Z-7015	Z-7020
	Part Number	XC7Z010	XC7Z015	XC7Z020
Processing System	Processor Core	Dual ARM® Cortex™-A9 MPCore™ with CoreSight™		
	Processor Extensions	NEON™ & Single / Double Precision Floating Point		
	Maximum Frequency	667 MHz (-1); 766 MHz (-2); 866 MHz (-3)		
	L1 Cache	32 KB Instruction, 32 KB Data per processor		
	L2 Cache	512 KB		
	On-Chip Memory	256 KB		
	External Memory Support ⁽¹⁾	DDR3, DDR3L, DDR2, LPDDR2		
	External Static Memory Support ⁽¹⁾	2x Quad-SPI, NAND, NOR		
	DMA Channels	8 (4 dedicated to Programmable Logic)		
	Peripherals ⁽¹⁾	2x UART, 2x CAN 2.0B, 2x I2C, 2x SPI, 4x 32b GPIO		
	Peripherals w/ built-in DMA ⁽¹⁾	2x USB 2.0 (OTG), 2x Tri-mode Gigabit Ethernet, 2x SC		
	Security ⁽²⁾	RSA Authentication, and AES and SHA 256-bit		
Processing System to Programmable Logic Interface Ports (Primary Interfaces & Interrupts Only)		2x AXI 32b Master 2x AXI 32-bit Slave 4x AXI 64-bit/32-bit Memory AXI 64-bit ACP 16 Interrupts		

Programozható logika (PL)

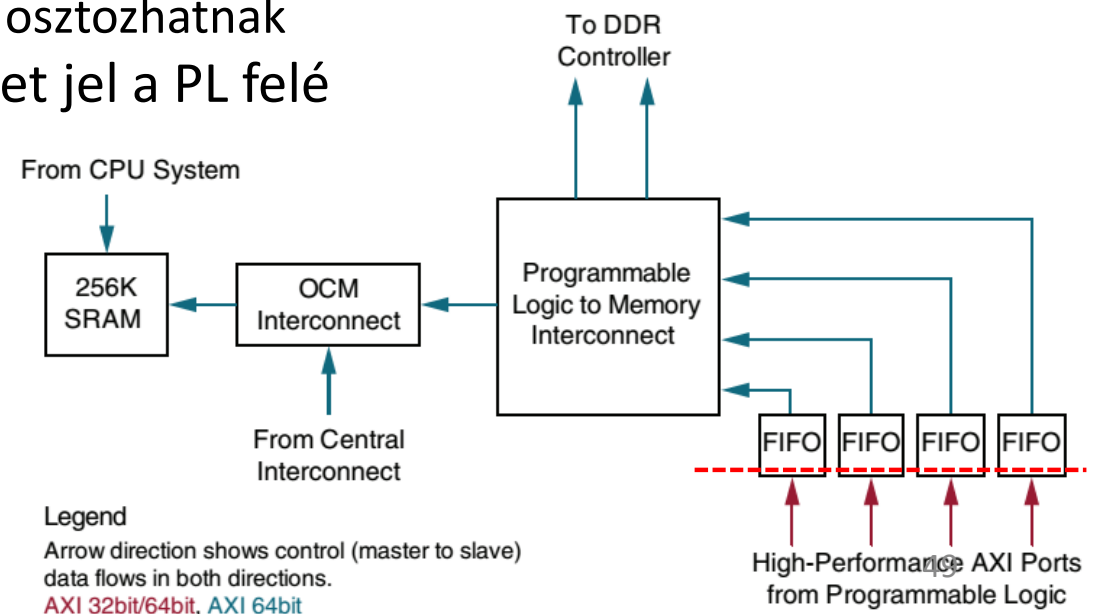


	Device Name	Z-7010
	Part Number	XC7Z010
Programmable Logic	Xilinx 7 Series Programmable Logic Equivalent	Artix®-7 FPGA
	Programmable Logic Cells (Approximate ASIC Gates) ⁽³⁾	28K Logic Cells (~430K)
	Look-Up Tables (LUTs)	17,600
	Flip-Flops	35,200
	Extensible Block RAM (# 36 Kb Blocks)	240 KB (60)
	Programmable DSP Slices (18x25 MACCs)	80
	Peak DSP Performance (Symmetric FIR)	100 GMACs
	PCI Express® (Root Complex or Endpoint)	—
	Analog Mixed Signal (AMS) / XADC	2x 12 bit, MSPS ADCs
	Security ⁽²⁾	AES and SHA 256b for

PS-PL összeköttetések



- AMBA AXI interfész nagysebességű elsődleges adatkommunikációhoz
 - 2 db 32-bites AXI master
 - 2 db 32-bites AXI slave
 - 4 db konfigurálható 32/64-bites pufferelt (FIFO) AXI slave interfész (OCM ill. DDR memória hozzáféréssel)
 - 1 db 64-bites AXI slave (ACP) – CPU memória koherencia (L1, L2)
- DMA, és megszakítás kezelő (GIC)
 - Processzor „event” busz
 - PL periféria megszakítása - a PS-ben lévő GIC felé
 - 4 DMA-csatorna (PL)
- EMIO: külső periféria I/O
 - PS oldali perifériák a PL lábakon is osztozhatnak
- 4 db PS órajel kimenet, 4 db PS reset jel a PL felé
- Konfiguráció
 - JTAG
 - XADC
 - Processor Configuration Access Port (PCAP)

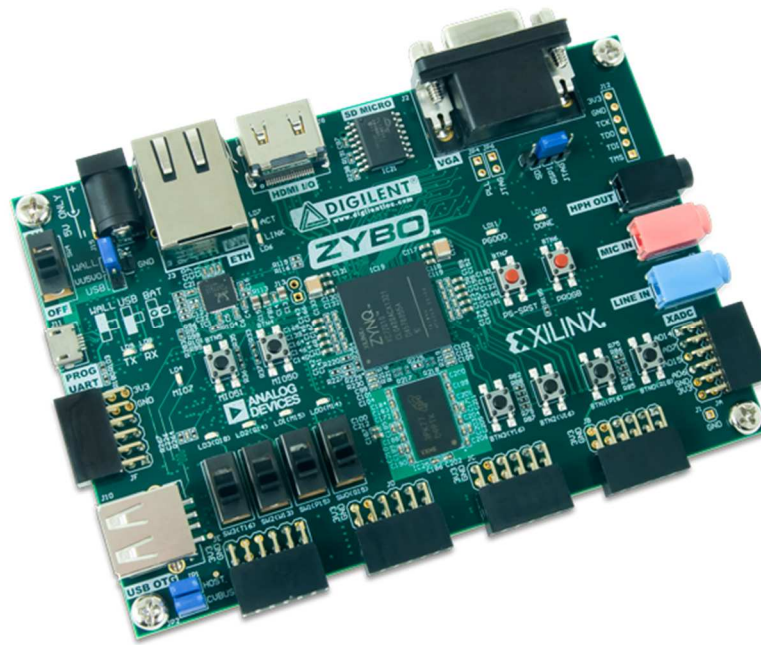


PL részei (7010)



A Zynq-7010 a **Xilinx Artix-7-es** FPGA családon alapuló architektúra.

- **CLB:**
 - 8 db LUT-6 / CLB: Logika és elosztott memória is lehet
 - Memória LUT: 64x1 bit RAM, vagy 32x2-bit RAM, vagy SRL
 - 16 db FFs / CLB
 - 2x4-bit láncolható összeadók (pl. FA)
- **BRAM** (36 Kbit):
 - Dual-port, max. 72-bit széles, vagy duál 18Kb BRAM
 - Programozható dedikált FIFO-ként
 - ECC támogatás (72-bites Hamming kódú kódszó)
- **DSP** szorzó: (előjeles, 25x18-bites)
 - 48-bit adder/accumulator
- Programozható **IOB-k**: számos szabvány (1.2-3.3V)
- Órajel-kezelés: **PLL/MMCM** – 2-2 db
- **XADC**: 2 db 12-bites A/D konverter, 1 MSPS
 - on-chip feszültség és hőmérséklet mérés, 17 differenciális input csatorna
 - www.xilinx.com/support/documentation/user_guides/ug480_7Series_XADC.pdf



Használt fejlesztő hardverek rövid bemutatása

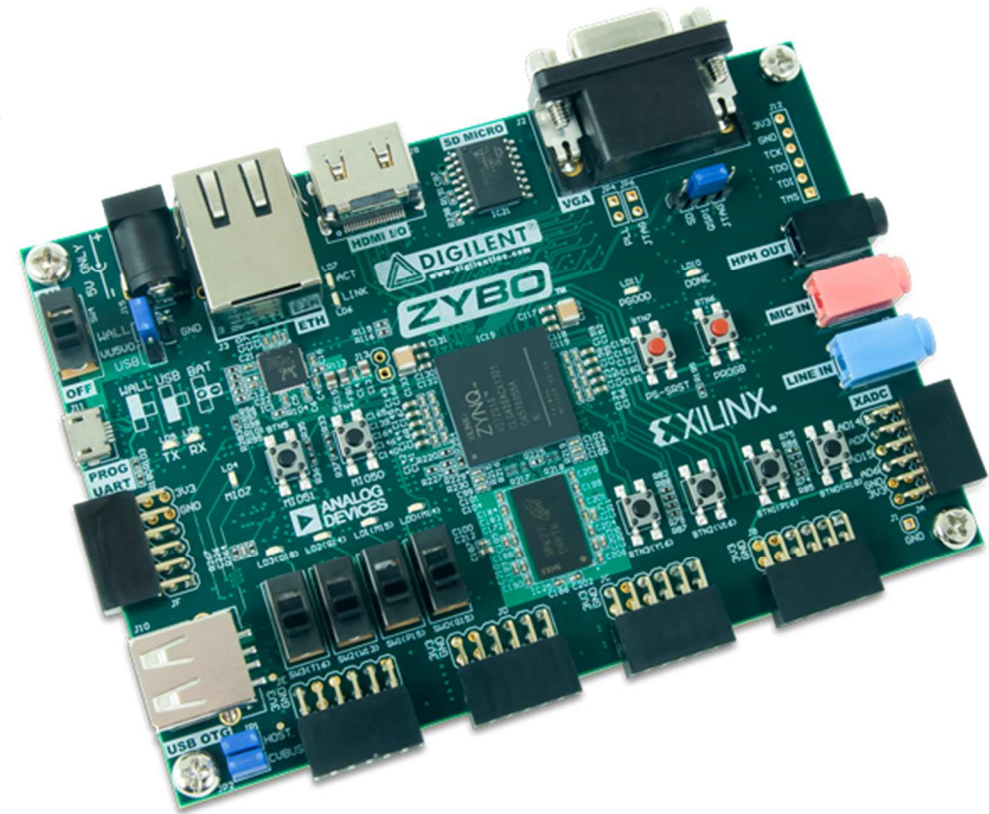
 **DIGILENT®** **ZYBO (ZYNQ BOARD)**
A National Instruments Company

-  ZYBO FPGA-s kártya hivatalos oldala:
www.digilentinc.com/Products/Detail.cfm?NavPath=2,400,1198&Prod=ZYBO
-  ZYBO Reference Manual (pdf):
www.digilentinc.com/Data/Products/ZYBO/ZYBO_RM_B_V6.pdf
-  ZYBO Reference Manual (online):
<https://reference.digilentinc.com/reference/programmable-logic/zybo/start>
-  ZYBO Schematic (kártya PCB és kapcsolási rajzok)
http://www.digilentinc.com/Data/Products/ZYBO/ZYBO_sch_B_V2.pdf
-  The ZYNQ book (reg.):
<http://www.zynqbook.com/>

Digilent ZYBO fejlesztő kártya

ZYBO™ Zynq FPGA fejlesztő kártya

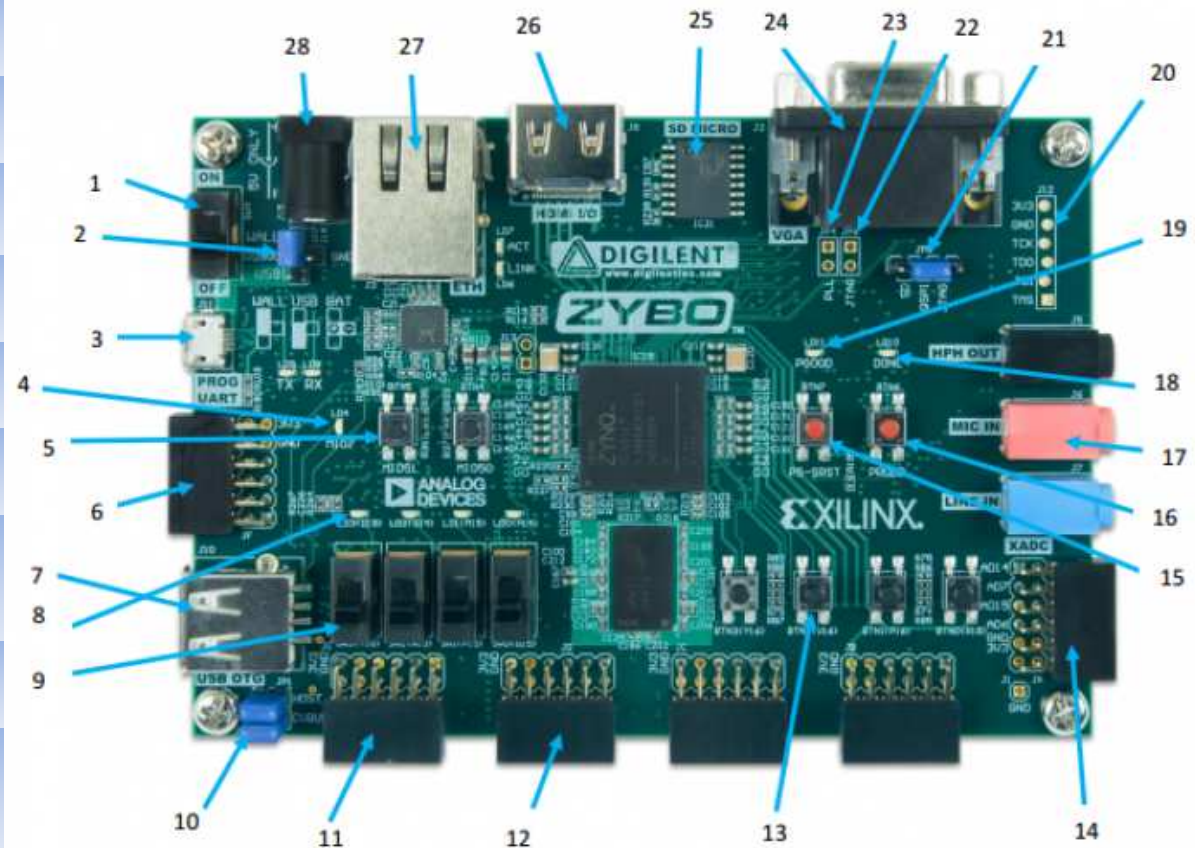
- *Xilinx Zynq-7000 (Z-7010)*
 - 650 MHz dual ARM Cortex-A9 magok (PS)
 - 8-csatornás DMA vezérlő (PS)
 - 1G ethernet, I2C, SPI, USB-OTG vezérlő (PS)
 - Artix-7 FPGA logika (PL), 54 muxed MIO pins (PL/PS)
 - 28Kbyte logikai cella, 240 Kbyte BRAM, 80 DSP szorzó(PL)
 - 12-bites, 1MSPS XADC (PL)
- 512 Mbyte DDR3 x32-bit (adatbusz), 1050Mbps sávszélességgel
- Tri-mode 10/100/1000 Ethernet PHY
- HDMI port: Dual role (source/sink)
- VGA port: 16-bites
- uSD kártya: OS tartalom tárolása
- OTG USB 2.0 (host és device)
- Audio codec
- 128Mbit x Serial Flash/QSPI (konfiguráció tárolási célokra)
- JTAG-USB programozhatóság, UART-USB vezérlő
- GPIO: 5 LED, 6 nyomógomb, 4 kapcsoló
- 4+1 PMOD csatlakozó (A/D átalakítóhoz)



Weblap: <http://digilentinc.com/Products/Detail.cfm?NavPath=2,400,1198&Prod=ZYBO>

Digilent ZYBO fejlesztő kártya

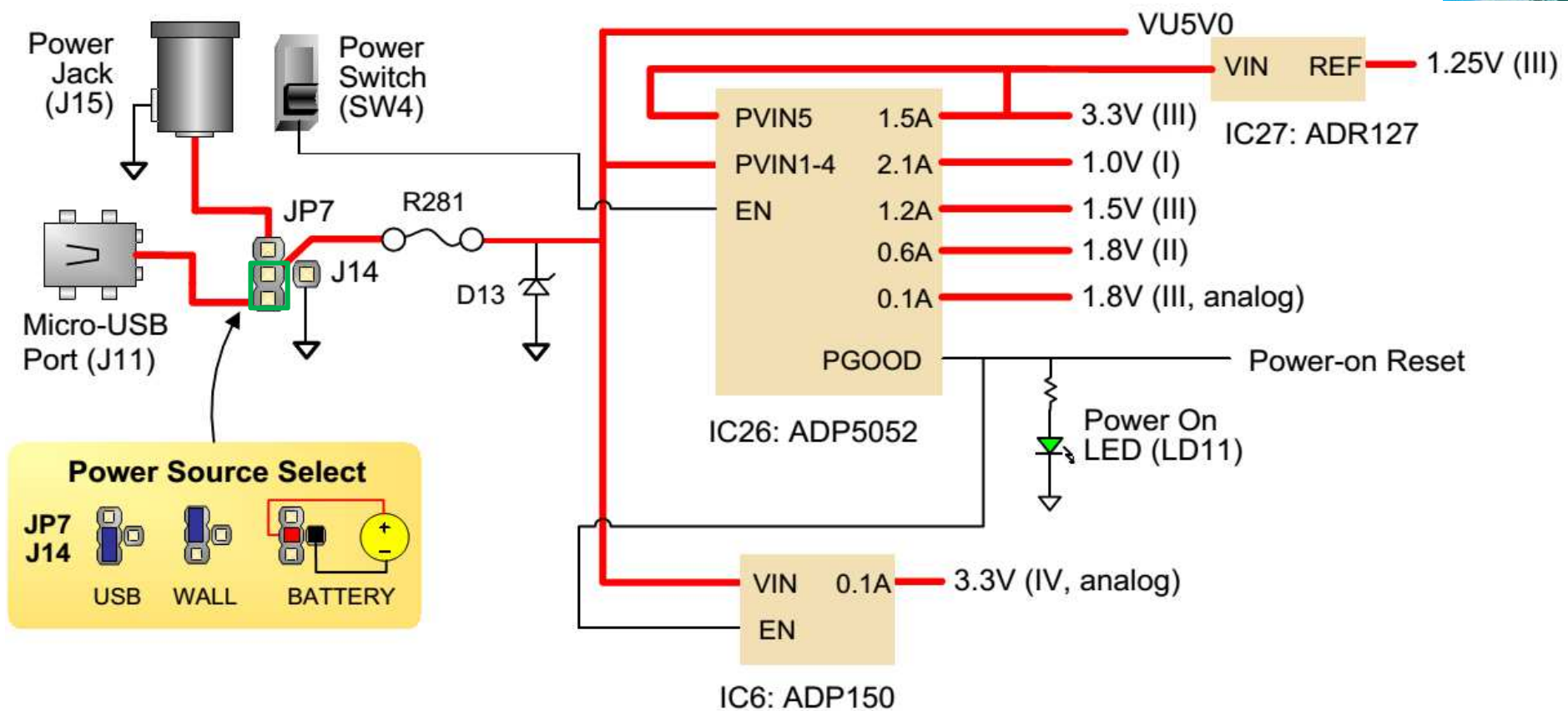
Callout	Component Description	Callout	Component Description
1	Power Switch	15	Processor Reset Pushbutton
2	Power Select Jumper and battery header	16	Logic configuration reset Pushbutton
3	Shared UART/JTAG USB port	17	Audio Codec Connectors
4	MIO LED	18	Logic Configuration Done LED
5	MIO Pushbuttons (2)	19	Board Power Good LED
6	MIO Pmod	20	JTAG Port for optional external cable
7	USB OTG Connectors	21	Programming Mode Jumper
8	Logic LEDs (4)	22	Independent JTAG Mode Enable Jumper
9	Logic Slide switches (4)	23	PLL Bypass Jumper
10	USB OTG Host/Device Select Jumpers	24	VGA connector
11	Standard Pmod	25	microSD connector (Reverse side)
12	High-speed Pmods (3)	26	HDMI Sink/Source Connector
13	Logic Pushbuttons (4)	27	Ethernet RJ45 Connector
14	XADC Pmod	28	Power Jack



Tápellátás

3-féle tápellátási lehetőség:

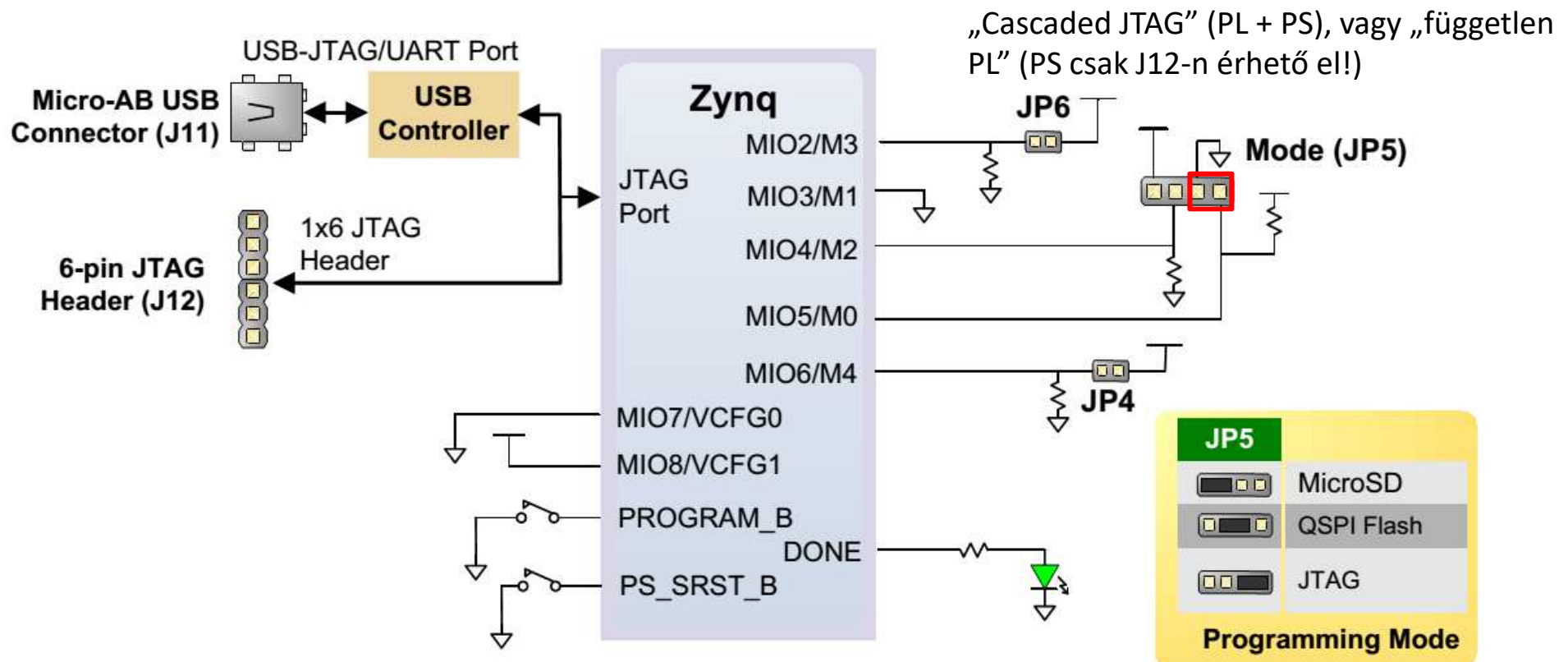
- mikro USB kábel (max 0.5A)
- fali csatlakozó (wall): AC adapter 5V/2.5A (Linux boot)
- elem (battery)



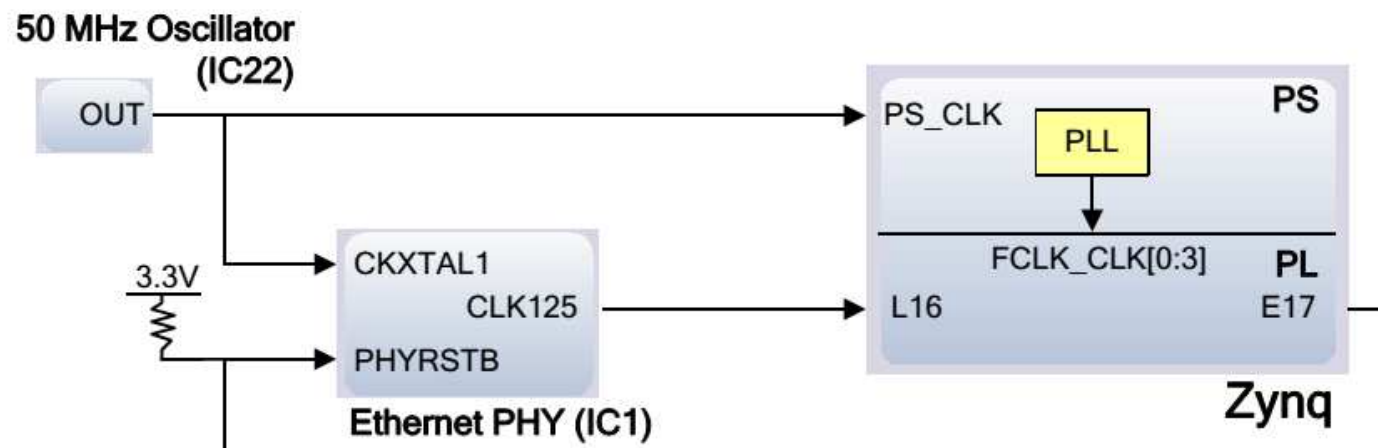
Programozhatóság

Konfigurálhatóság/Programozhatóság lehetőségei:

1. USB JTAG/UART interfész (J11),
2. JTAG-en keresztül (+ **debug** funkció J12 – nincs beültetve): ehhez kell egy Xilinx Platform USB programozó eszköz is,
3. microSD kártya (J4): pl. Linux boot image (3 fázisa van a betöltésnek *BootRom* -> *FSBL* -> *SSBL* / *sw. alkalmazás*) ,
4. QSPI soros Flash-en keresztül (PS/PL konfigurálhatóság)



- 50 MHz PS_CLK (PS alrendszer): PLL-el 4 különböző referencia órajel generálható
- 125 MHz ref.clk (PL alrendszer), független PS-től (L16 láb)
- ARM: max. 650 MHz órajel
- DDR3 Memória vezérlő: max. 525 MHz (1050 Mbps)



Programozható I/O perifériák

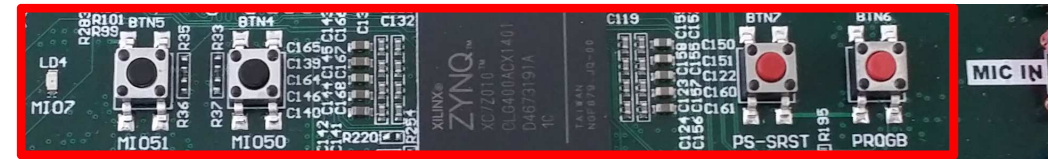
PS alrendszerben:

- Nyomógombok: 2 db
- LED: 1 db

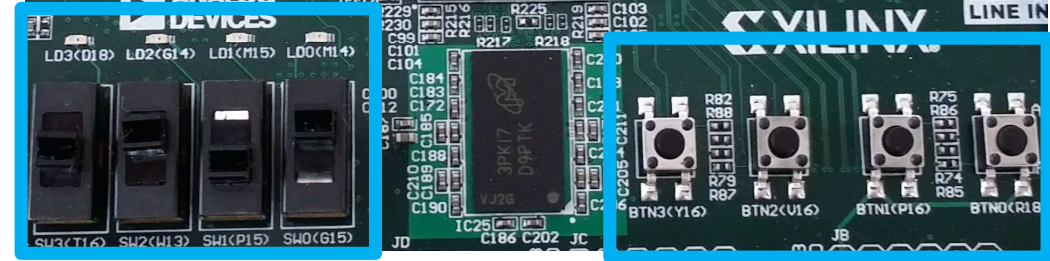
PL alrendszerben:

- Nyomógombok: 4 db
- Kapcsolók: 4 db
- LED-ek: 4 db

PS

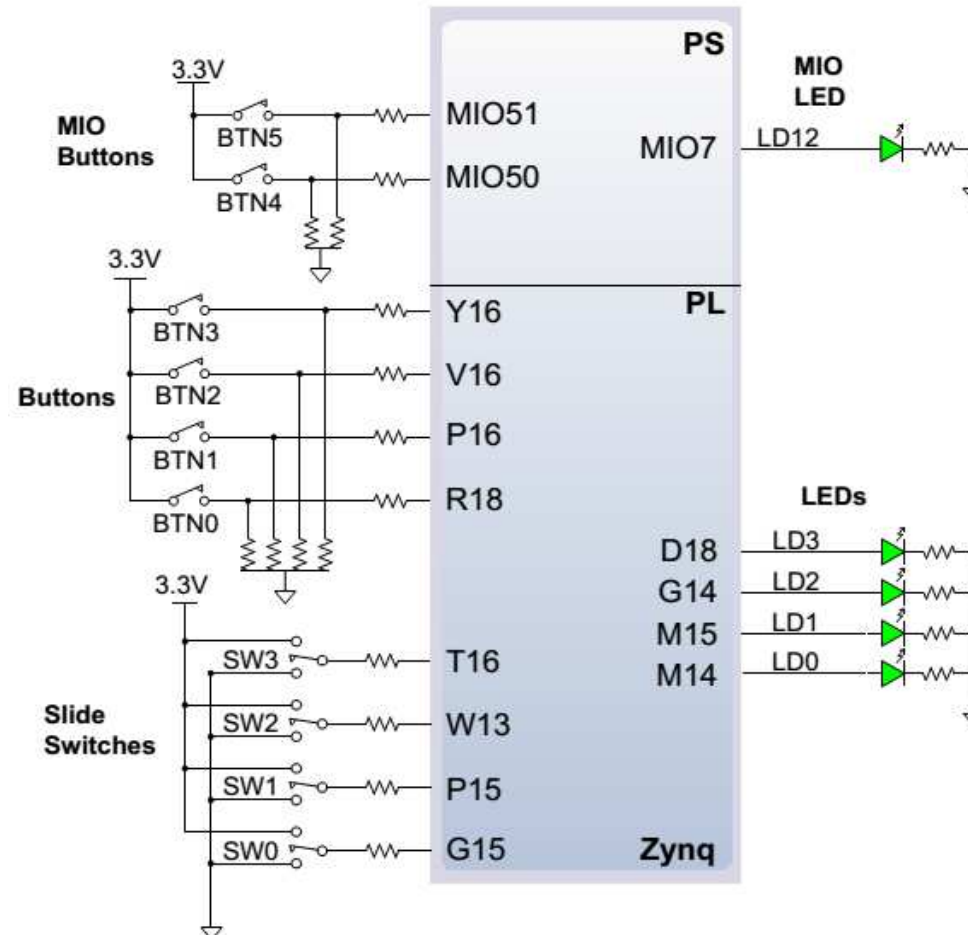


PL



Nyomógombok:
'0': inaktív
'1': aktív

2-állású Kapcsolók:
'0': inaktív
'1': aktív



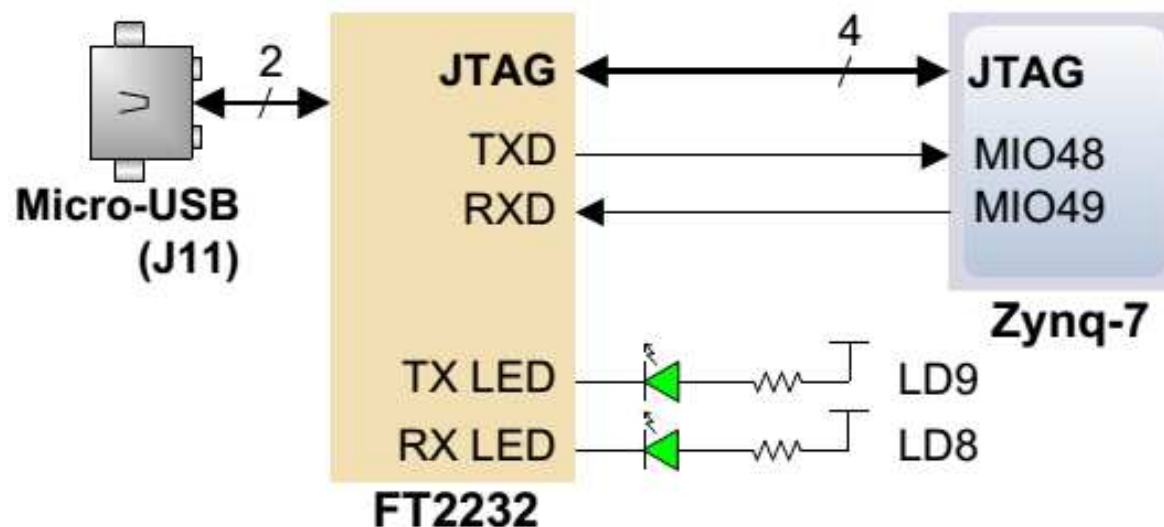
LED: (közös katódos)
'0': inaktív
'1': aktív

Külső memóriák

- Micron DDR3 SDDRAM (512 MByte):
MT41K128M16JT-125 DDR3 típus
 - 32 bites adatbusz
 - PS (ARM) oldali dedikált memória (hard-memory) vezérlő
- Spansion soros SPI Flash 128Mbit (16 MByte):
S25FL128S típus
 - 1x, 2x SPI-, 4x QSPI mód
 - max. 400 Mbit/s elérési sebesség (QSPI módban)
 - PS, PL oldal konfigurálhatósága

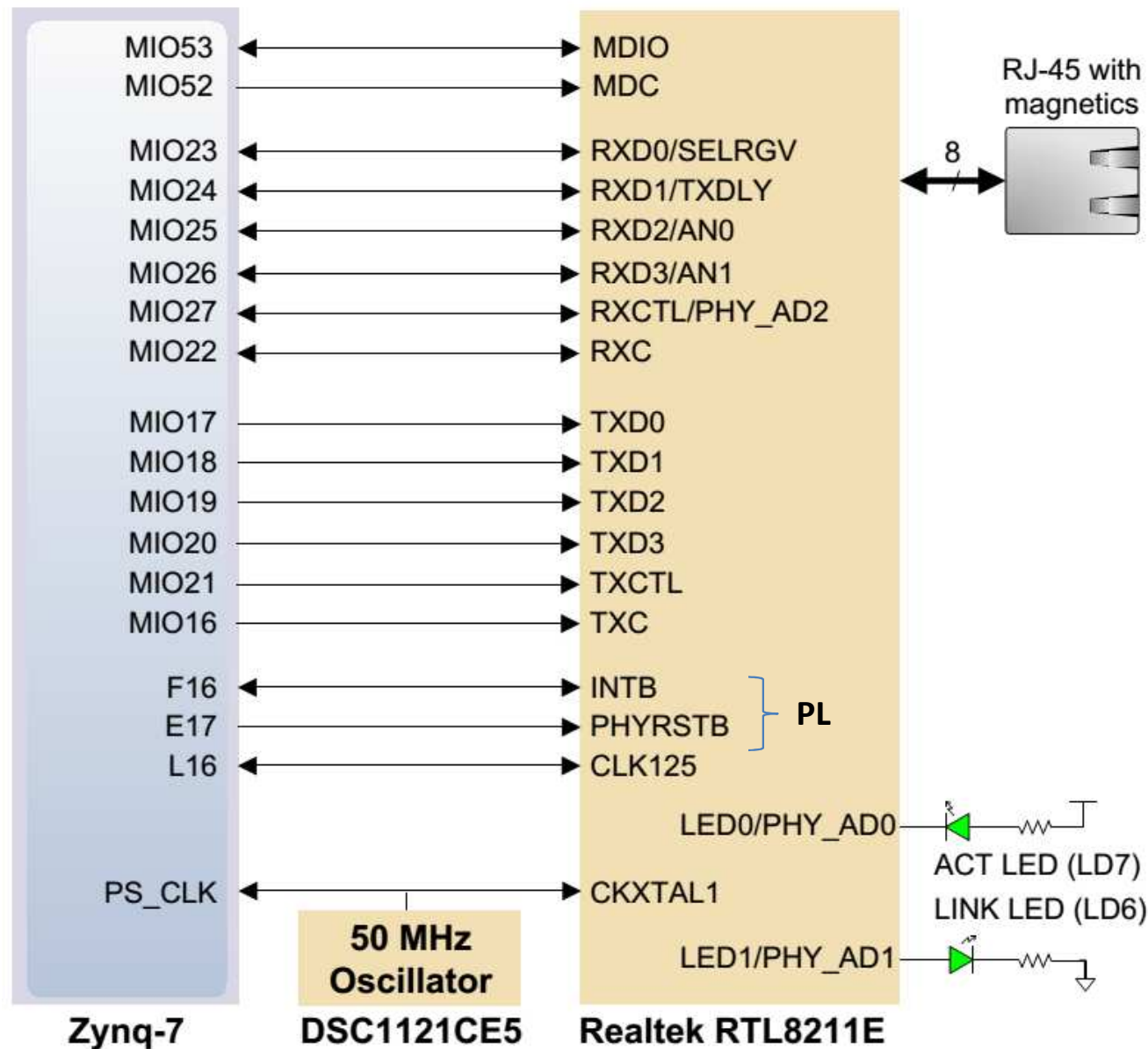
USB-UART

- FTDI USB-UART híd (J11): *FT2232HQ* típus
 - USB <-> soros UART csomag kontroll
 - Protokoll beállítások: 115.200 baud rate, 1 stop bit, no parity, 8-bit adat
 - PS oldali elérés: MIO 48-49
 - Két funkció egyben: JTAG-UART + tápellátás



Ethernet vezérlő

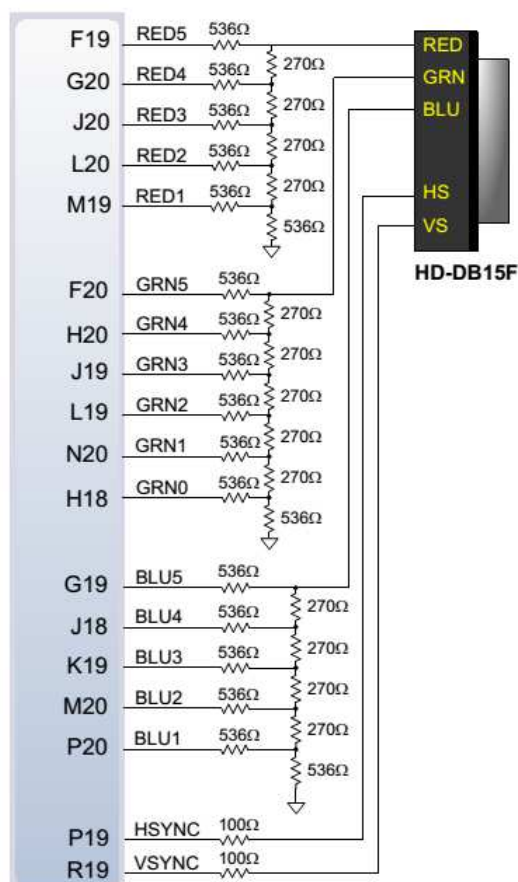
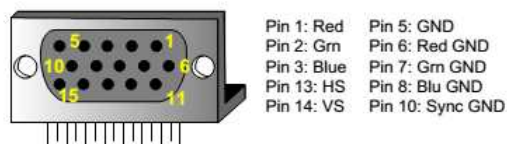
- Realtek RTL8211E-VL PHY : 10/100/1000 Mbit/s
 - RGMII átviteli mód, Gigabit Ethernet MAC



Video be-/kimenetek

- 1 VGA kimenet (16-bites színmélység: 5 Red, 6 Green, 5 Blue csatorna + HS, VS szinkronizációs jelek)
- 1 HDMI bemenet, vagy 1 HDMI kimenet (opcionálisan)

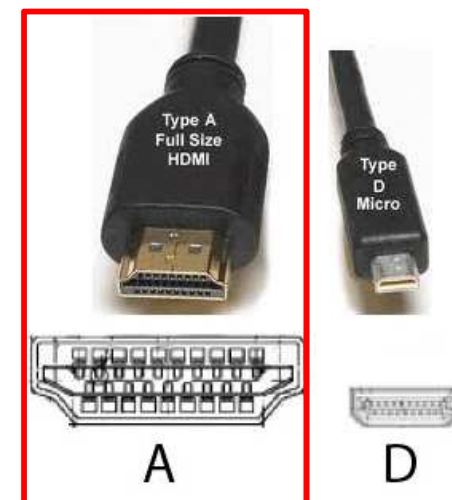
VGA



Zynq-7

HDMI

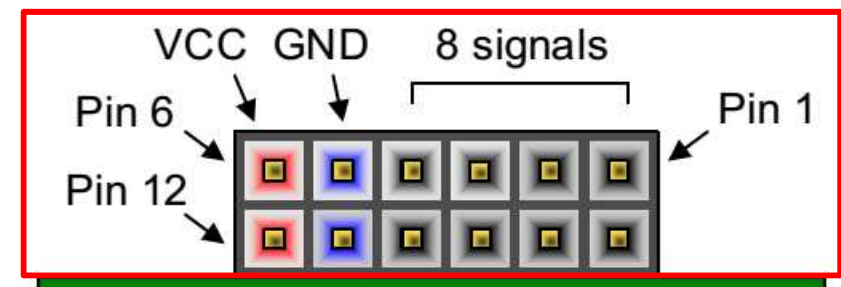
- Bemenetként, vagy kimenetként konfigurálható HDMI port (PL)
 - Source / Sink mód
 - HDMI-A konnektor
- HDMI/DVI jelek kódolásához, dekódolásához nincs külön videó chip a kártyán, azaz FPGA logikában kell megvalósítani!



- Analog Devices SSM2603 audio codec
 - 8-96 KHz mintavétel
 - Monó mikrofon bemenet
 - Sztereó line in bemenet
 - Sztereó kimenet (3.5 mm jack)
- PS alrendszerhez kapcsolódik I2C busz interfészen keresztül
 - De I2S protokollt használ az adatok továbbítására

 <http://www.analog.com/media/en/technical-documentation/data-sheets/SSM2603.pdf>

- Összesen 6 db PMOD csatlakozó (egyenként 12 láb: 2 táp + 2 föld + 8 adatjel)
 - Standard PMOD: PL alrendszer érheti el (4 db)
 - MIO PMOD: PS alrendszer érheti el (1 db)
 - Dual **A/D** (XADC*) PMOD: **PL** alrendszer használhatja (1 db):
 - 12 –bit A/D konverter, 1 MSPS



Bővítési lehetőség - PMOD

- Periféria modulok (Pmod), további bővítési lehetőségek

- Karakteres LCD, OLED, 7segLEG
- GPS vevő, WiFi, Bluetooth,
- Ethernet IF, USB-UART, RS232
- Joystick, Rotary Enc., Switches,
- SD Card, Serial Flash,
- A/D, D/A konverterek, H-hidak
- Gyorsulás-mérő, Perdület-mérő (Gyroscope),
- Hőmérséklet-mérő, ...stb.

vagy

- „3rd party” gyártók, esetleg egyedi tervezésű bővítőkártyák fejlesztése, és illesztése.

