

Mikroelektromechanikai rendszerek

Beágyazott rendszerek szoftverei. Operációs rendszerek és programozási nyelvek.

Oktató: Kajdocsi László

Iroda: Informatika Tanszék, A602

Email: kajdocsi.laszlo@sze.hu



Oktató: Tüű-Szabó Boldizsár

Iroda: Informatika Tanszék, B606/A

Email: tuu.szabo.boldizsar@sze.hu



Beágyazott rendszerek

- **Beágyazott rendszer** (Embedded System): a (számítógépes) hardver- és szoftverelemeknek 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

Beágyazott rendszerek

Alkalmazási területek:

- **Autóipari alkalmazások:** beágyazott elektronikus vezérlők
 - Biztonságkritikus: központi elektronikai vezérlő (ECU), motorvezérlés, fékrásegítő, sebességváltó, blokkolásgátló vezérlés (ABS), kipörgésgátló (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

Beágyazott rendszerek

Alkalmazási területek:

- **Gyógyászati berendezések**
 - Orvosi képfeldolgozás
 - Jelmonitorozás (PET, MRI, CT)
- **Hálózati/ telekommunikációs rendszerek** (modem, router stb.)
- **IoT:** Intelligens, vagy smart rendszerek
- **Háztartási gépek, ill. fogyasztói elektronika**

Beágyazott rendszerek

Általános követelmények:

- **Dedikált funkció**
 - Jól körülhatárolt (alkalmazás specifikus) funkció(k) támogatása
- **Szigorú követelmények**
 - Alacsony költség (Cost)
 - Gazdaságosság (Economy) - lehetőleg minimális alkatrészből épüljön fel
 - Gyors működés (Speed)
 - Alacsony disszipáció (Power)
- **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)**

Beágyazott rendszerek

Alapkövetelmények:

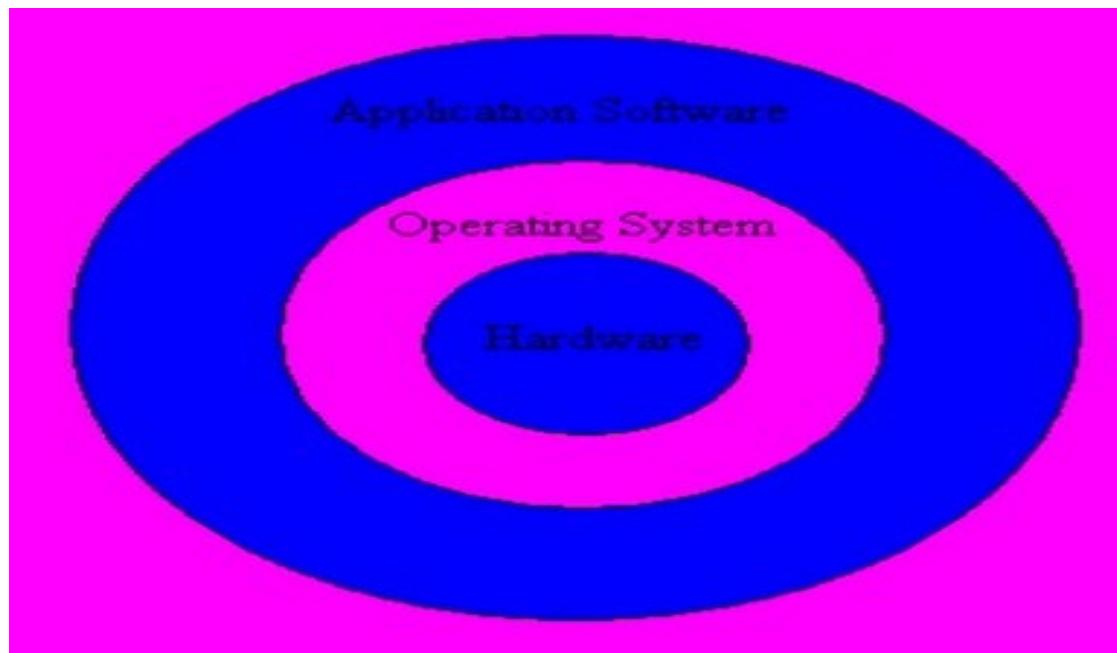
- **Idő:** Egy bekövetkező esemény kezelését a beágyazott rendszer egy meghatározott időn belül kezdje el.
- **Biztonság:** olyan rendszer vezérlése, amely hibás működés esetén egészségkárosodás, és komoly anyagi kár nélkül kezeli a bekövetkező eseményt.

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

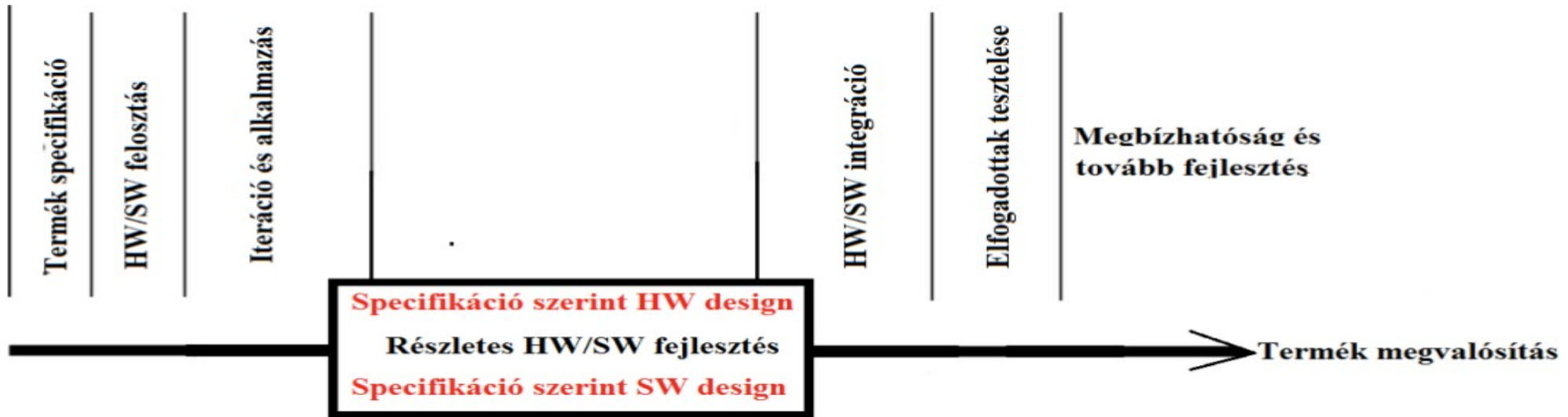
Beágyazott rendszerek

Beágyazott rendszerek komponensei:



Beágyazott rendszerek

Beágyazott rendszerek tervezése:



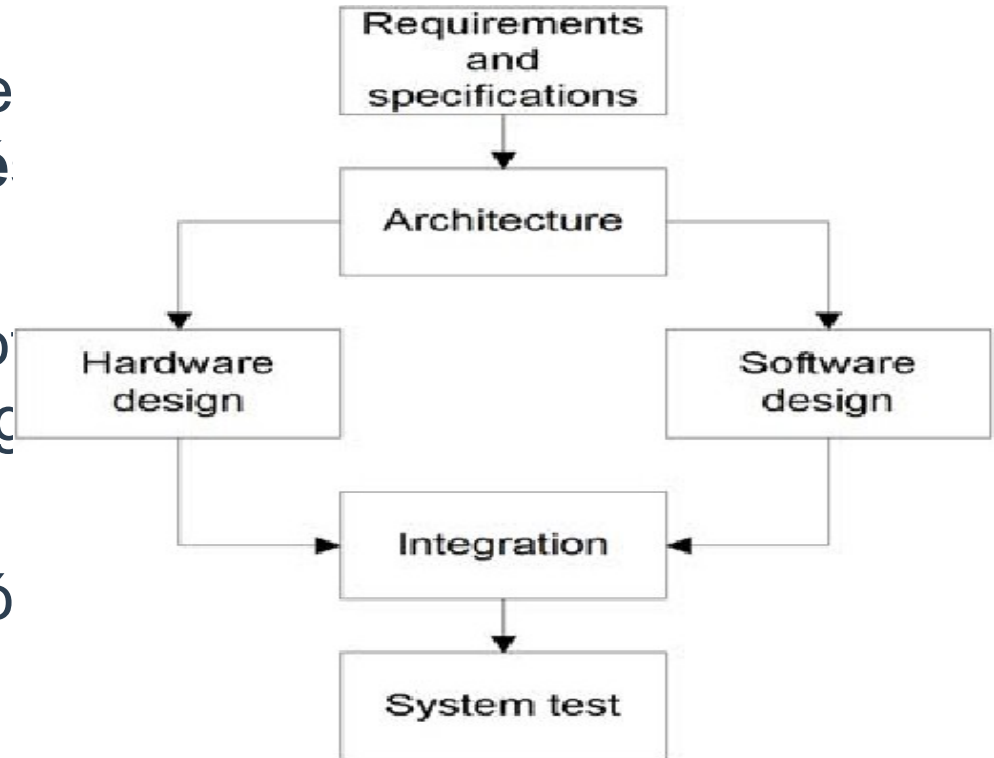
Beágyazott rendszerek

Hardver/szoftver partícionálás:

- Annak eldöntése, hogy mely funkcionalitást érdemes hardverben, melyet szoftverben megvalósítani
- Figyelembe vehető költségek: hardver (pl. terület), szoftver (pl. memória), kommunikáció
- Különféle korlátok és optimalizálási célok: real-time, terület, ár, energia
- Probléma legtöbb változata erősen NP-nehéz

Beágyazott rendszerek

- Hardver és szoftver együttes tervezése, **szimulációja és verifikációja**
- Nem kell várni a beágyazott szoftver tervezésével addig, amíg elkészül a hardver
- A tervezés magas absztrakció szinten történhet



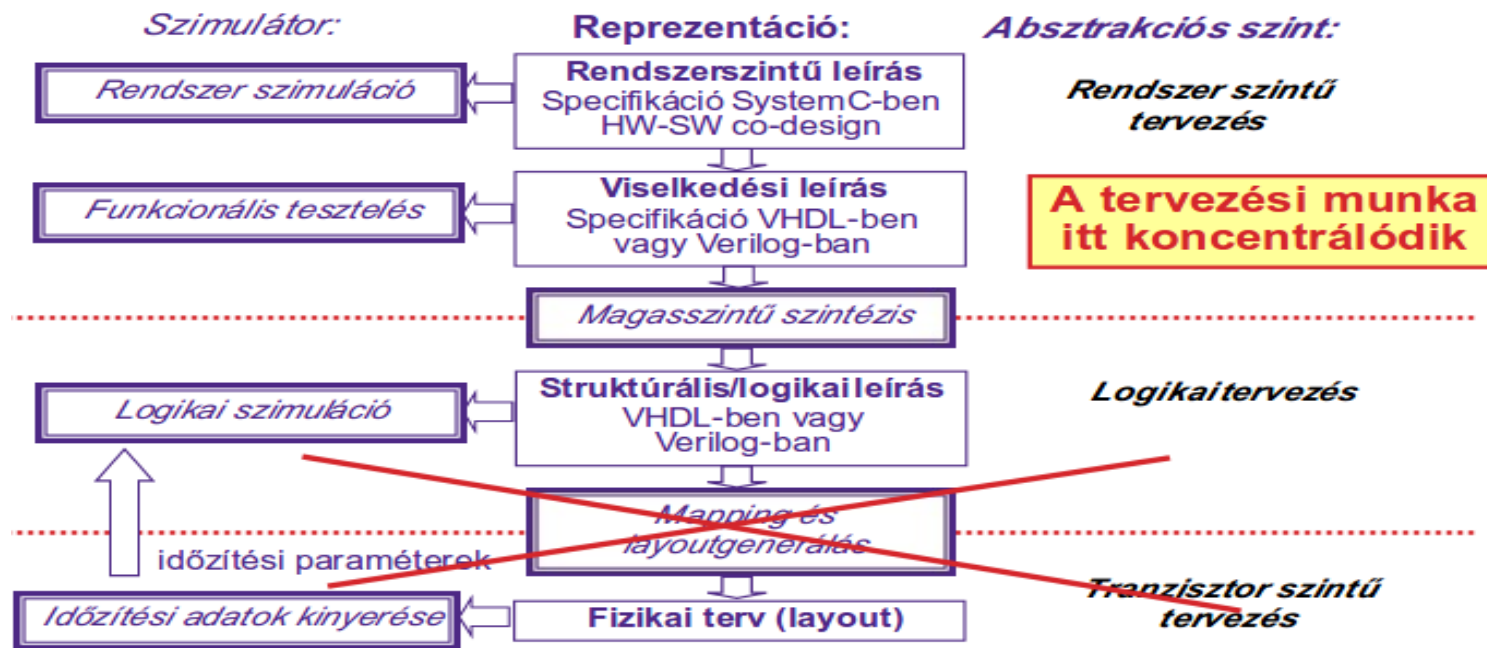
Beágyazott rendszerek

Hardver és szoftver együttes tervezése, szimulációja és verifikációja:

- Magas szintű absztrakciós leírással könnyebben átlátható legyen a terv
- Hibákat gyorsabban meg lehessen találni
- Optimalizálni lehessen a hardver/szoftver partícionálást
- Csökkenjen a piacra kerülés átfutási ideje
- Csökkenjenek a fejlesztési költségek

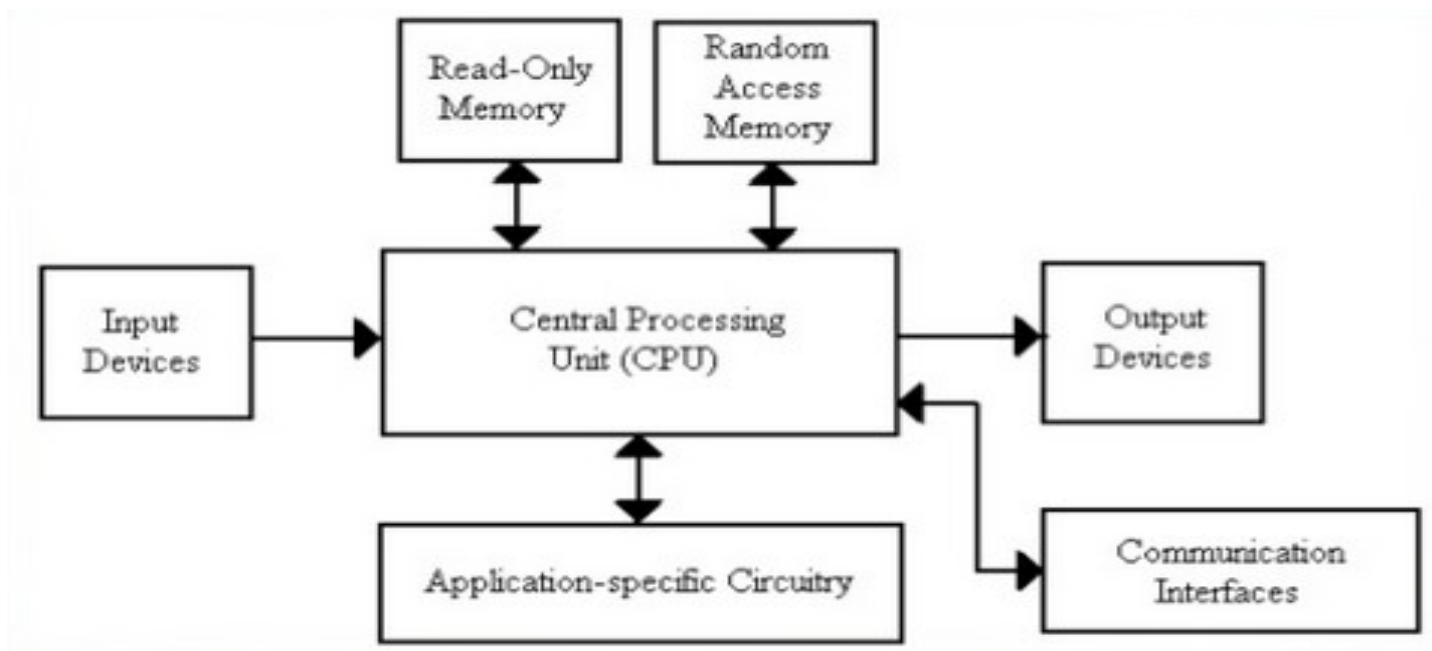
Beágyazott rendszerek

FPGA-s tervezés folyamata:



Beágyazott rendszerek

Hardverarchitektúra:



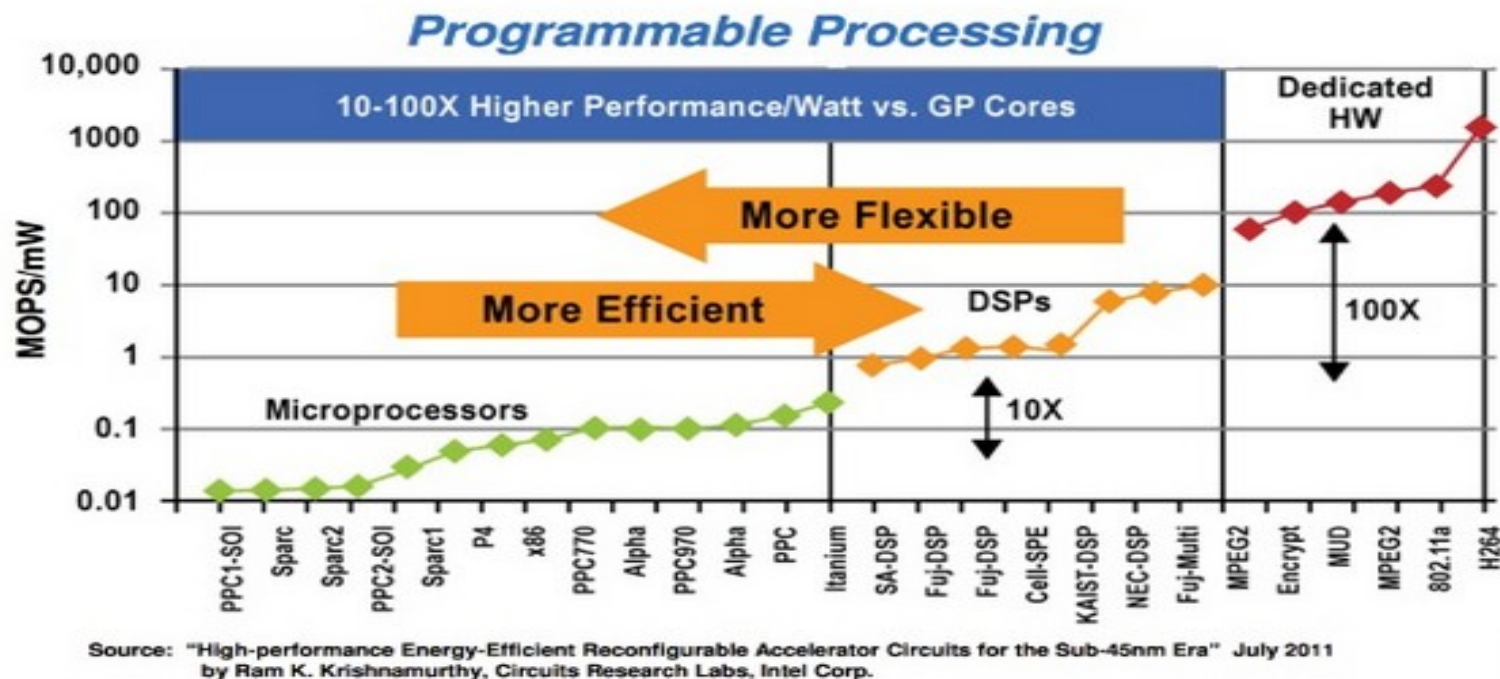
Beágyazott rendszerek

A központi egység megvalósításai:

- Mikrovezérlő alapú rendszerek (microcontroller)
- DSP processzor alapú rendszerek
- ASIC technológián alapuló rendszerek
- FPGA technológián (programozható logikai áramkörök) alapuló rendszerek

Beágyazott rendszerek

Hardver:



Beágyazott rendszerek

Hardver:

	General purpose		Application-specific	Single purpose
Programmable in	Software	Hardware	Software	-
Performance	low	medium	medium	high
Energy Efficiency	low	medium	medium	high
Feature Flexibility	very high	medium	low	very low
Per unit price savings	low	low	medium	very high
Example	Microprocessor	PLD / FPGA	DSP	ASIC / ASSP

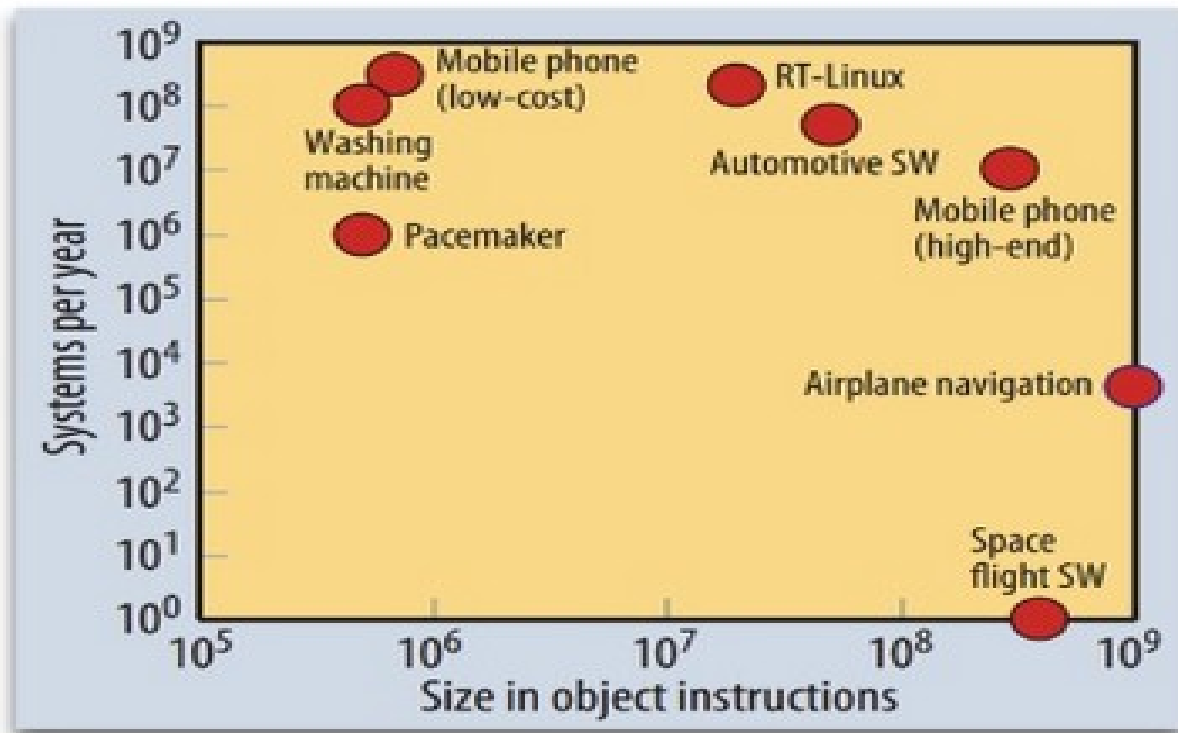
Beágyazott rendszerek

Szoftver:

- más jellegű követelmények lépnek fel, mint az általános célra épített számítógépek szoftvereinél:
- hardver szempontból sokkal kisebb erőforrással (kisebb RAM, gyengébb órajelű processzor, stb.) kell beérni
- sokkal hibatűrőbbnek kell lenni
- követelmény lehet a valósidejű futás is

Beágyazott rendszerek

Szoftver:



Beágyazott rendszerek

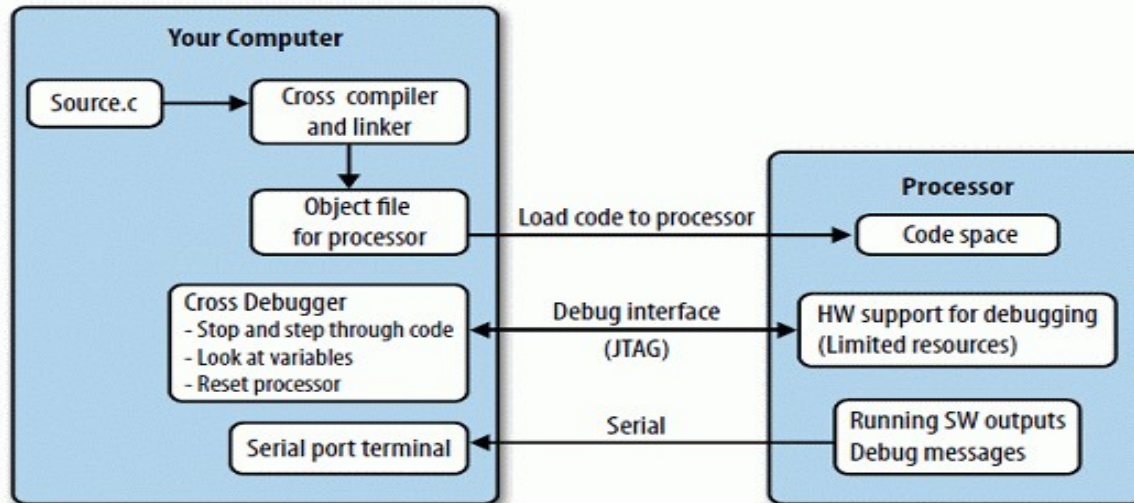
Szoftver:

- Fejlesztés általános célú számítógépeken
- Fordítás keresztfordító program (cross compiler) segítségével beágyazott rendszeren futó képes kódra (target)
- Cross compiler: olyan fordító program, amely az őt futtató platformtól eltérő platformon futó kód generálására is képes.

Beágyazott rendszerek

Szoftver:

- A beágyazott szoftver fejlesztésének és nyomkövetésének blokksémája



Beágyazott rendszerek

Programozási nyelvek:

- Gépi kódban való programozás
 - A processzor számára önmagában értelmezhető bináris adat
 - Manapság igen ritkán alkalmazzák
- Assembly
 - Rövid szöveges utasítások
 - Még mostanság is sok helyen alkalmazzák mikrokontrollerek, vagy akár processzorok programozására, ha egy utasítás nagy hatékonyságú elvégzése kulcs fontosságú, illetve esetlegesen az adott utasításnak nincs magasabb programnyelvű implementációja
 - Komoly hátránya, hogy a processzorhoz, pontosabban az adott architektúrához kötöttek

Beágyazott rendszerek

Programozási nyelvek:

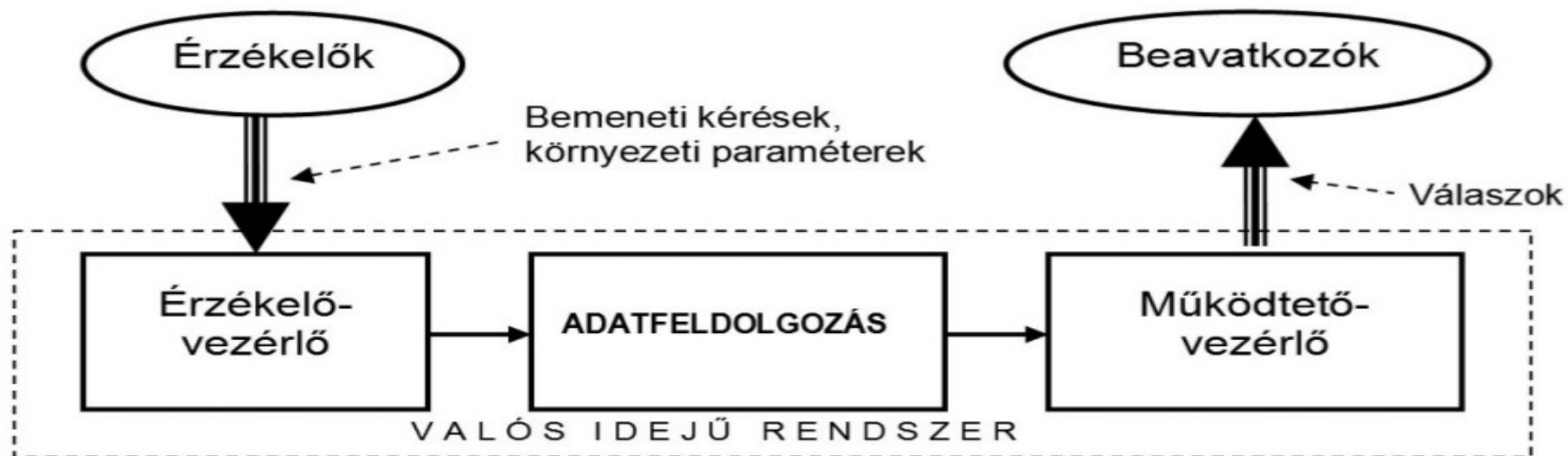
- C nyelv
 - Egyik legelterjedtebb, a kisebb kontrollerekre talán a leginkább alkalmazott programozási nyelv
- **Objektumorientált programozási nyelvek:** C++, Java
 - Elsősorban összetett programkódok esetén szokták alkalmazni, ahol fontos a modularitás.
 - Kisméretű kontrollerek esetén többnyire nem kellően hatékonyak.

Beágyazott rendszerek

Grafikus programozási nyelvek:

- Pl. Matlab/Simulink, NI LabVIEW, stb.:
 - Nem programkódot kell írni, hanem grafikus módon, blokkok segítségével kerül összeállításra a program
- Hátrányok:
 - ingyenes verzió ritkán érhető el ezen programokból
 - csak adott típusú kontrollereket támogatnak

Valós idejű rendszerek



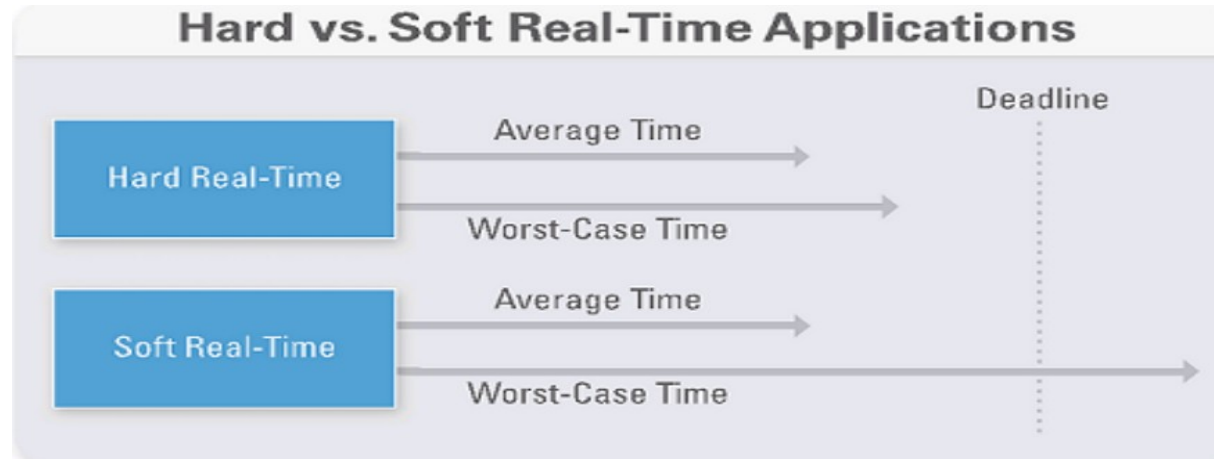
Valós idejű rendszerek

A legegyszerűbb felépítésűeket kivéve, a beágyazott rendszerek tartalmazznak valós idejű operációs rendszert (Real Time Operating System – RTOS)

- **Válaszidő:** Egy esemény által kiváltott időzítő (trigger) impulzus és az eseményt lekezelő program indulása között eltelt idő.
- **Határidő teljesített:** Ha egy eseményt a megadott válaszidőnkön belül elkezdte a rendszer feldolgozni.
- **Hibás rendszerviselkedés:** Ha a válaszidők az előírt időhatáron kívül vannak.

Valós idejű rendszerek

- **Erősen valós idejű rendszerek (Hard real-time):** a műveletek érvénytelenek, ha az eredmények nem állnak elő a meghatározott időn belül. Minden határidősértés elfogadhatatlan, és megengedhetetlen.
- **Gyengén valós idejű rendszerek (soft real-time):** kritikus folyamatokat a rendszer mindössze nagyobb prioritással dolgozza fel.



Biztonságkritikus rendszerek

Példák: nukleáris erőművek, vasút, autóipari alkalmazások, légi közlekedés
szigorú rendeletek szabályozzák a biztonsági követelményeket

Szoftvernél **aktív redundancia**: Eltérő tervezésű redundáns modulok szükségesek

Variánsok: azonos specifikáció, de

- eltérő algoritmus, adatstruktúrák
- más fejlesztési környezet, programnyelv
- elszigetelt fejlesztés

Beágyazott operációs rendszerek

Többféle csoportosítás lehetséges:

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

Tulajdonságaik alapján léteznek:

- **Valós idejű operációs rendszerek** - garantálnak valamilyen előírt időkorláton belüli válaszidőt (pl. pilóta nélküli repülőgép)
- **Nagy megbízhatóságú operációs rendszerek** – magas rendelkezésre állás (pl. űrszonda)
- **Konfigurálható operációs rendszerek** – funkciók kiválasztása, testre szabása

Beágyazott operációs rendszerek

Beágyazott processzorok valós-idejű operációs rendszerei (RTOS):

- Linux
- Android
- Micrium uC/OS
- QNX
- RTLinux
- Windriver VxWorks (RT)
- Symbian
- Windows Embedded
- Windows CE

Valós idejű operációs rendszerek

A berendezés összetettségétől függ, hogy a valós idejű operációs rendszer mely elemei lesznek beépítve. A RTOS leggyakrabban ezeket az elemeket foglalja magában:

- **valós idejű óra** (real time clock) – a periodikus folyamatok időzítéséhez;
- **megszakításkezelő** (interrupt handler) – az aperiodikus események feldolgozására;
- **ütemező** (scheduler) – ez a komponens választ ki egyet a futtatható folyamatok közül egyet a végrehajtásra;
- **erőforrás-kezelő** (resource handler) – memóriát és processzoridőt rendel a futtatható folyamathoz;
- **elosztó** (distributor) – folyamat futtatását kezdi meg.

Valós idejű operációs rendszerek

Az összetettebb berendezések – pl. a vezérlő és távközlési berendezések – valós idejű operációs rendszerei további elemeket is tartalmazhatnak, úgy mint:

- **háttértár kezelő**
- **hiba felismerő és – jelző mechanizmus**
- **dinamikus alkalmazás-újrakonfiguráló**



**SZÉCHENYI
EGYETEM**
UNIVERSITY OF GYŐR



Köszönöm a figyelmet!