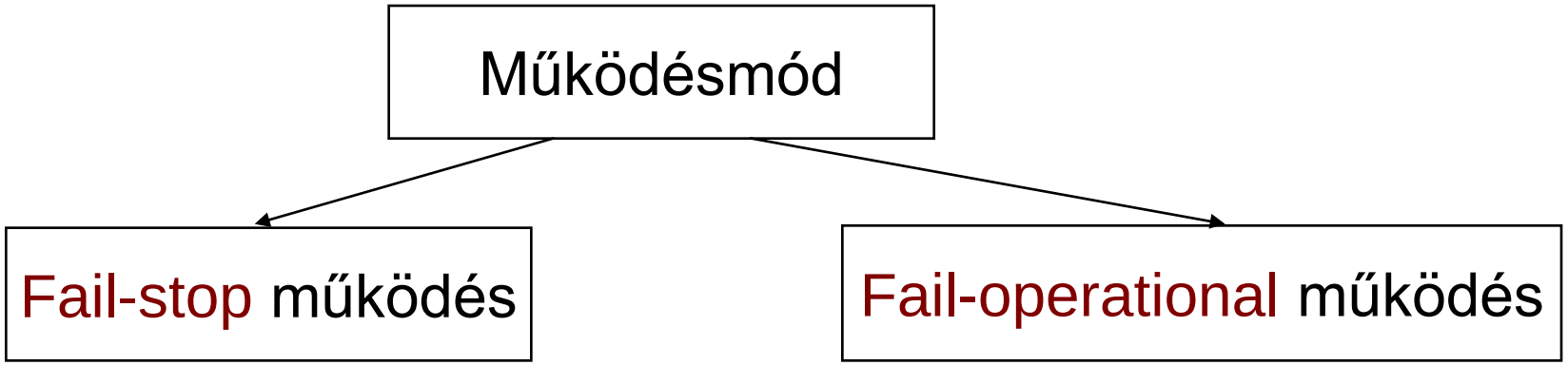
Architektúra tervezési példák: Architektúrák biztonságkritikus rendszerekben

Majzik István
majzik@mit.bme.hu

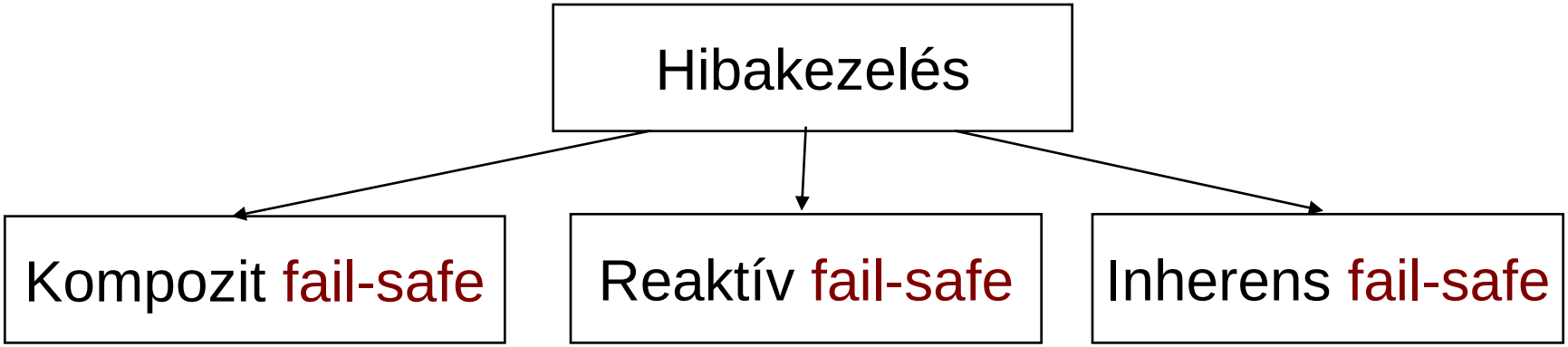
Biztonságos állapotok



- A megállás (lekapcsolás) biztonságos állapot
- Detektált hiba esetén le kell állítani a rendszert
- Hibadetektálás a kritikus feladat

- A megállás (lekapcsolás) nem biztonságos állapot
- Detektált hiba esetén is szükséges szolgáltatás
  - teljes, vagy
  - csökkentett (degradált)
- Hibatűrés szükséges

Általános biztonsági alapelvek



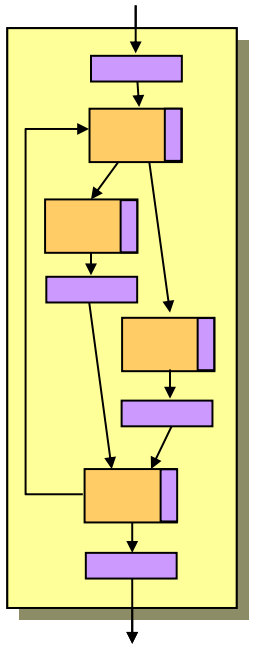
- Minden funkciót megvalósít legalább 2 független komponens
- A továbblépéshez (többségi) egyetértés szükséges
- Több, mint 2 komponens esetén hibatűrés (leszavazás)

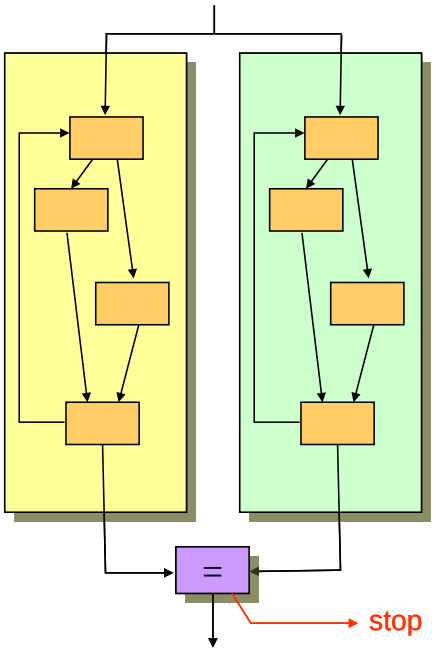
- Minden funkció mellé rendelhető független hibadetektálás
- A detektált hiba hatása negálható

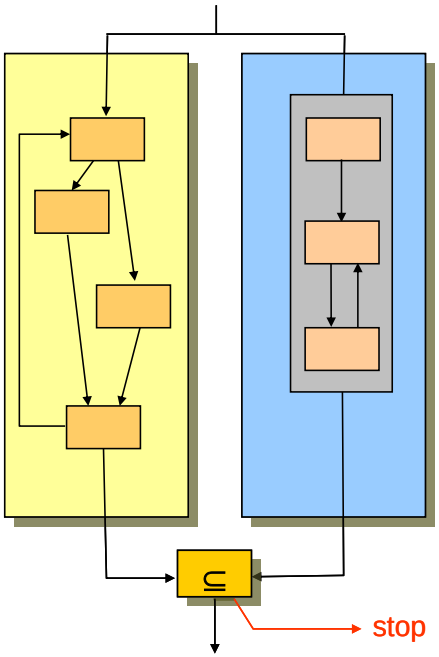
- Minden hibamód veszélytelen
- „Természeténél fogva biztonságos”

Jellegzetes megoldások fail-stop rendszerekhez

- Áttekintés:
  - Egycsatornás feldolgozás öntesztttel
  - Két- vagy többcsatornás feldolgozás komparálással
  - Kétcsatornás feldolgozás független ellenőrzéssel

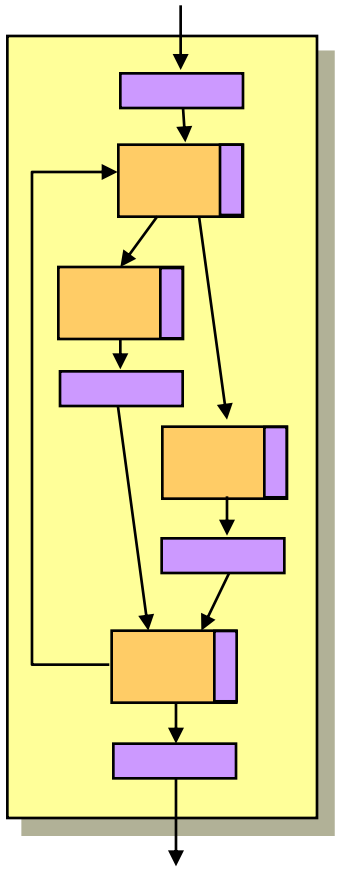






I. Egycsatornás feldolgozás öntesztel

- Egy feldolgozási folyamat
- Ütemezett hardver öntesztek
  - Induláskor részletes önteszt
  - Futás közben: Lappangó állandósult hibák detektálása
  - Ajánlott többféle módon
- Rendszeres szoftver önellenőrzés
  - Végrehajtási utak ellenőrzése
  - Alkalmazásfüggő hibadetektálás
- Hátrányok:
  - Önteszt hibafedése korlátos
  - Hibakezelés megvalósítása kérdéses (pl. lekapcsolás)



Példa: SAFEDMI mozdonyvezetői kezelőfelület

Mozdony-vezető

DMI

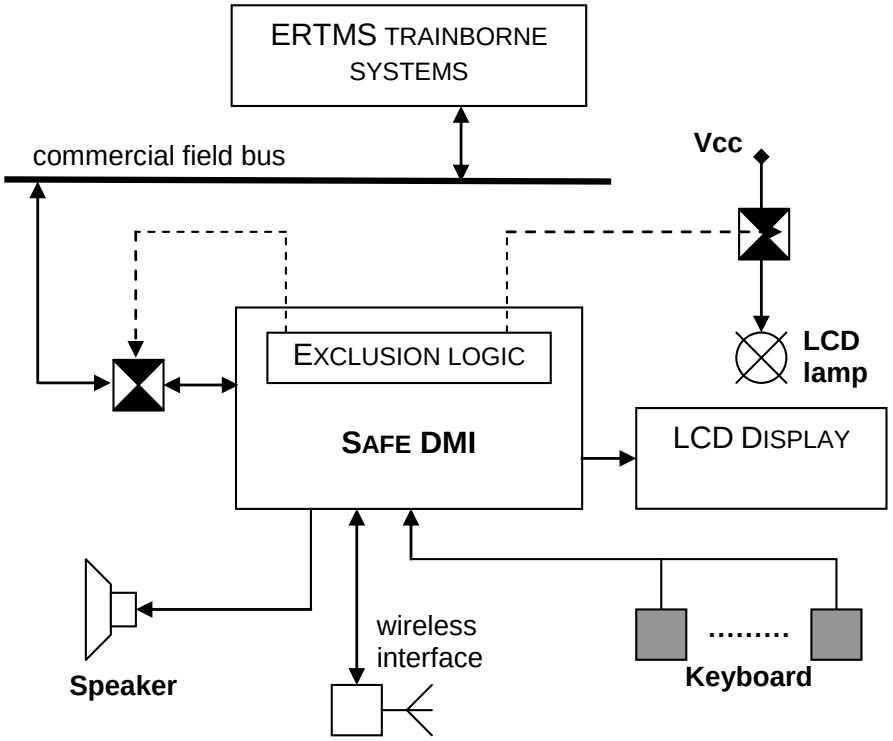
EVC
European Vital Computer

Maintenance Centre

- Safety Integrity Level 2
  - Információ megjelenítés
  - Parancs feldolgozás
  - EVC kommunikáció
- Biztonságos vezeték nélküli kommunikáció
  - Konfiguráció
  - Diagnosztika
  - Szoftver frissítés

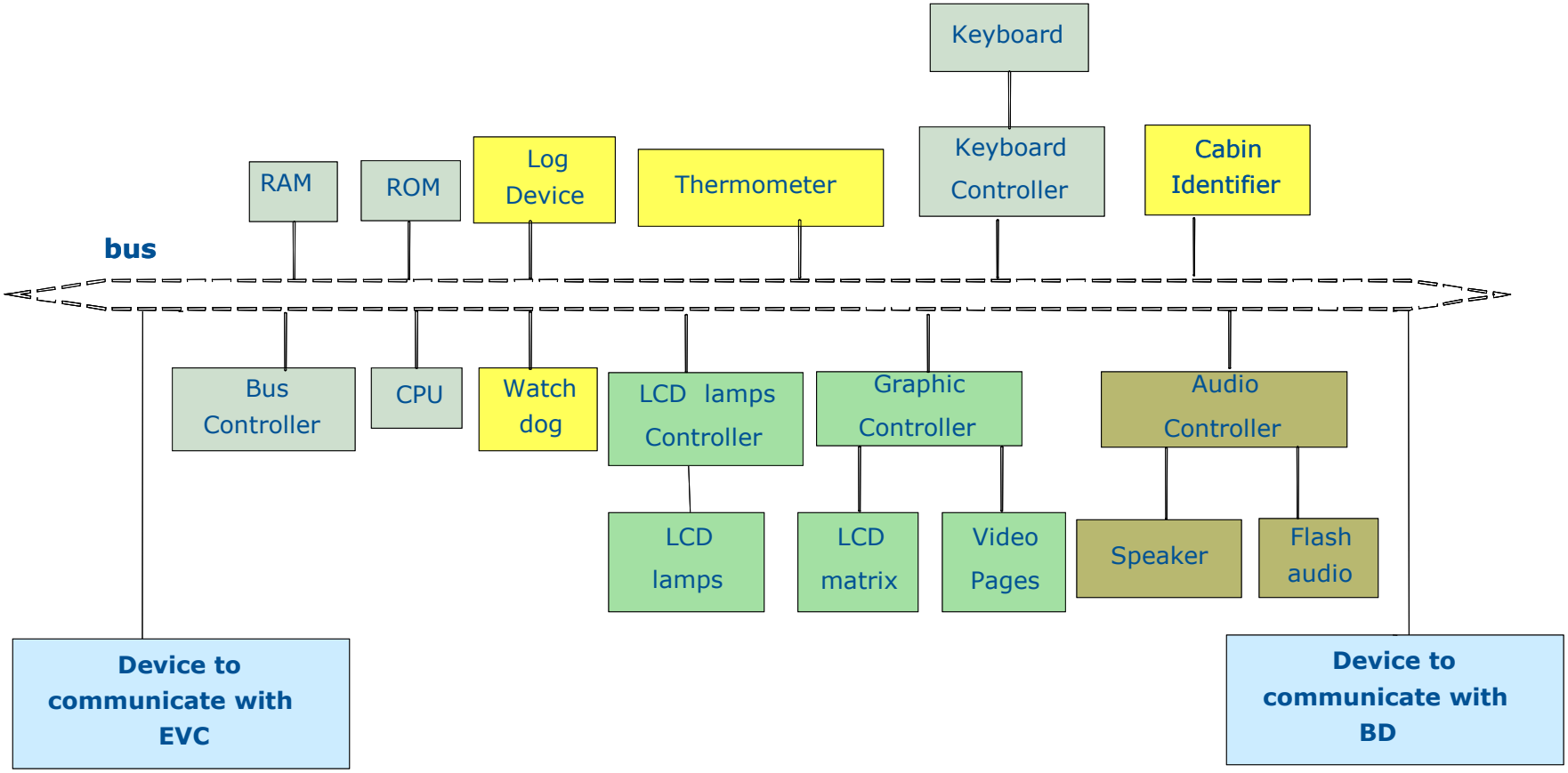
Példa: SAFEDMI hardver architektúra

- Reaktív fail-safety (hibadetektálás és hibakezelés)
- Generikus hardver komponensek
- A hibadetektálás és hibakezelés megvalósítása szoftver alapú megoldásokkal



Példa: SAFEDMI hardver architektúra

Komponensek:

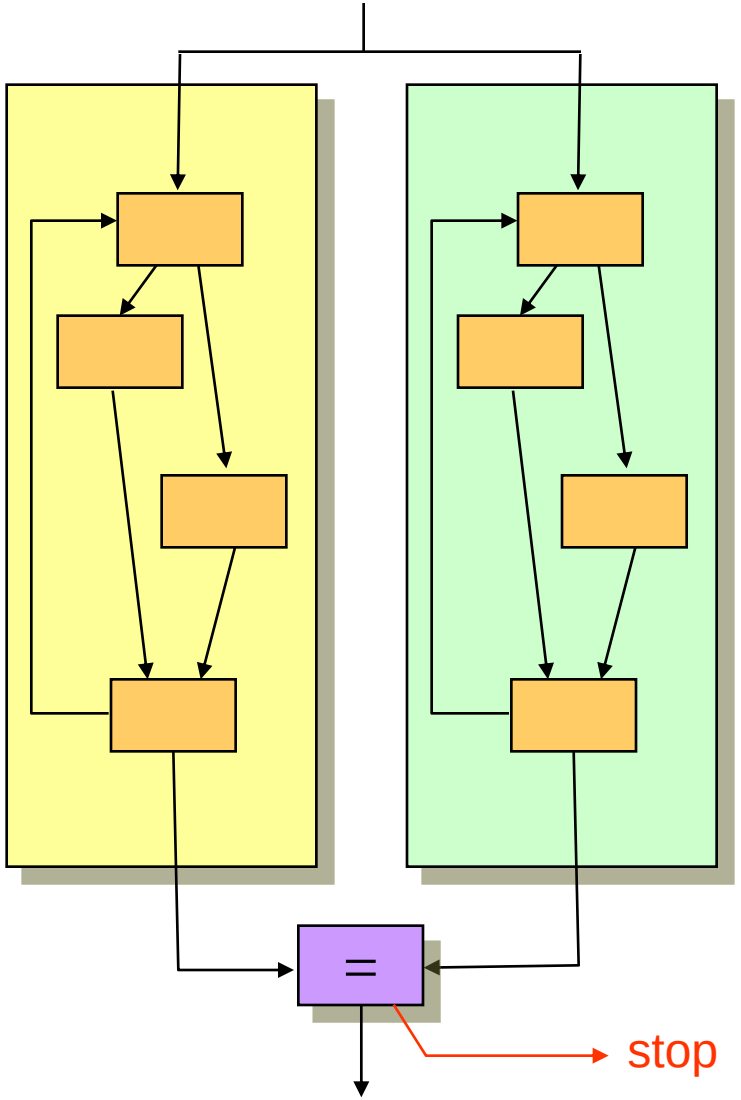


Példa: SAFEDMI hibadetektálási technikák

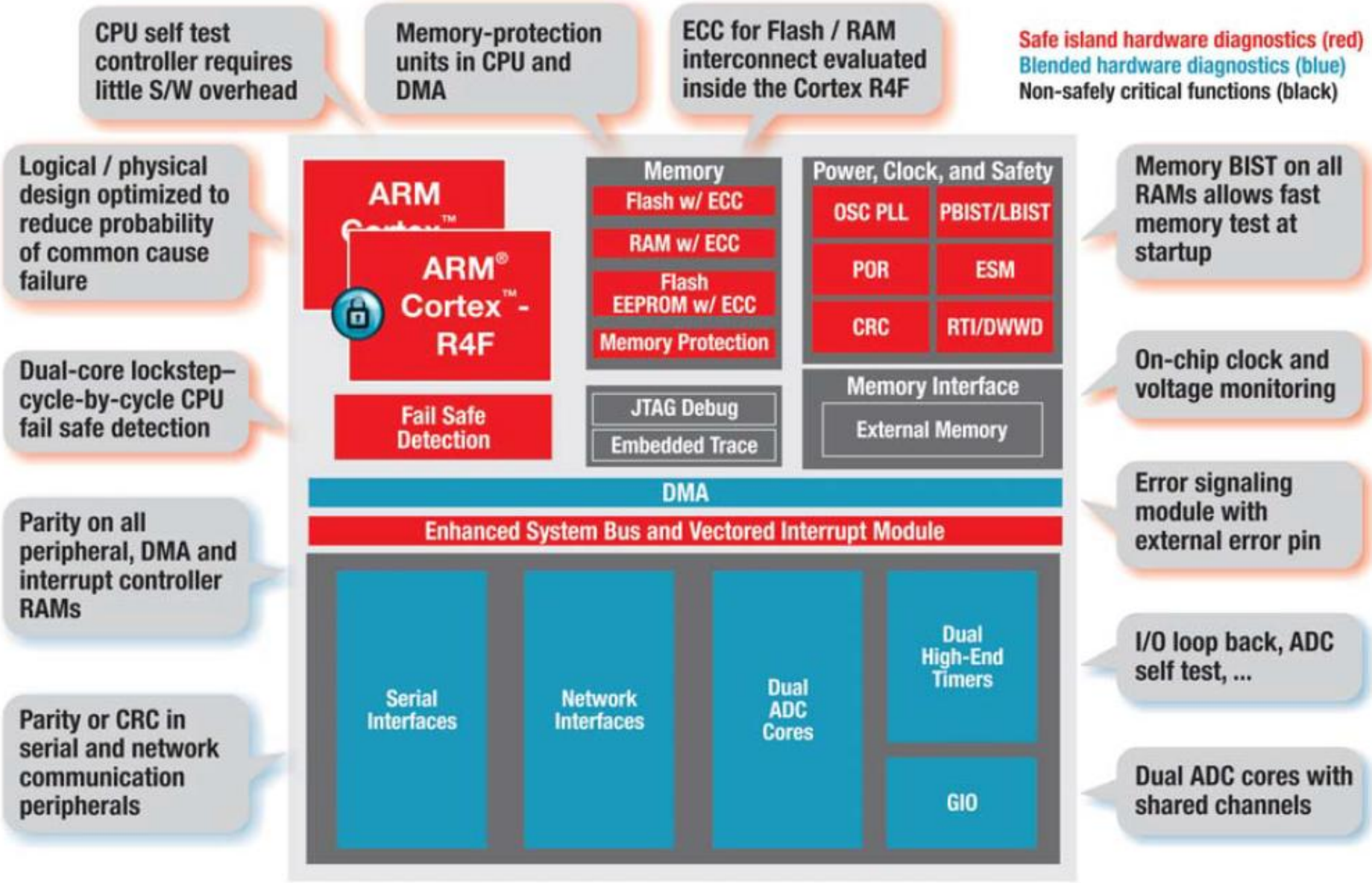
- Elindulás:
  - Részletes öntesztelés az állandósult hibák detektálására
- Működés közben:
  - Periodikus önteszt: Kisebb erőforrásigényű technikákkal
  - On-line ellenőrzések
    - Kommunikáció, konfiguráció esetén: Adat elfogadhatósági / hihetőségi vizsgálatok
    - Vezérlés-orientált funkciókra: Vezérlési folyam monitorozása
    - Adat-orientált funkciókra: Duplikált számítás és összehasonlítás
- Kommunikáció:
  - Adathiba detektálása és javítása: Erős hibajavító kódolás
  - (Adat)védelem: Titkosítás
  - Biztonság: Adott hibagyakoriság fölött kapcsolat bontás

II. Két- vagy többcsatornás feldolgozás

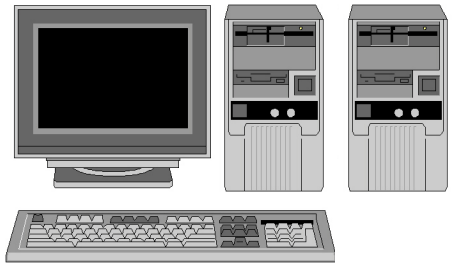
- Két vagy több feldolgozási csatorna
  - Közös bemenet
  - Kimenetek komparálása
  - Eltérés esetén leállás
- Nagy hibafedés
- Komparátor kritikus elem
  - De egyszerű!
  - Kiváltása kódolt feldolgozással
- Hátrányok:
  - Közös eredetű hiba?
  - Hosszú lappangási idő



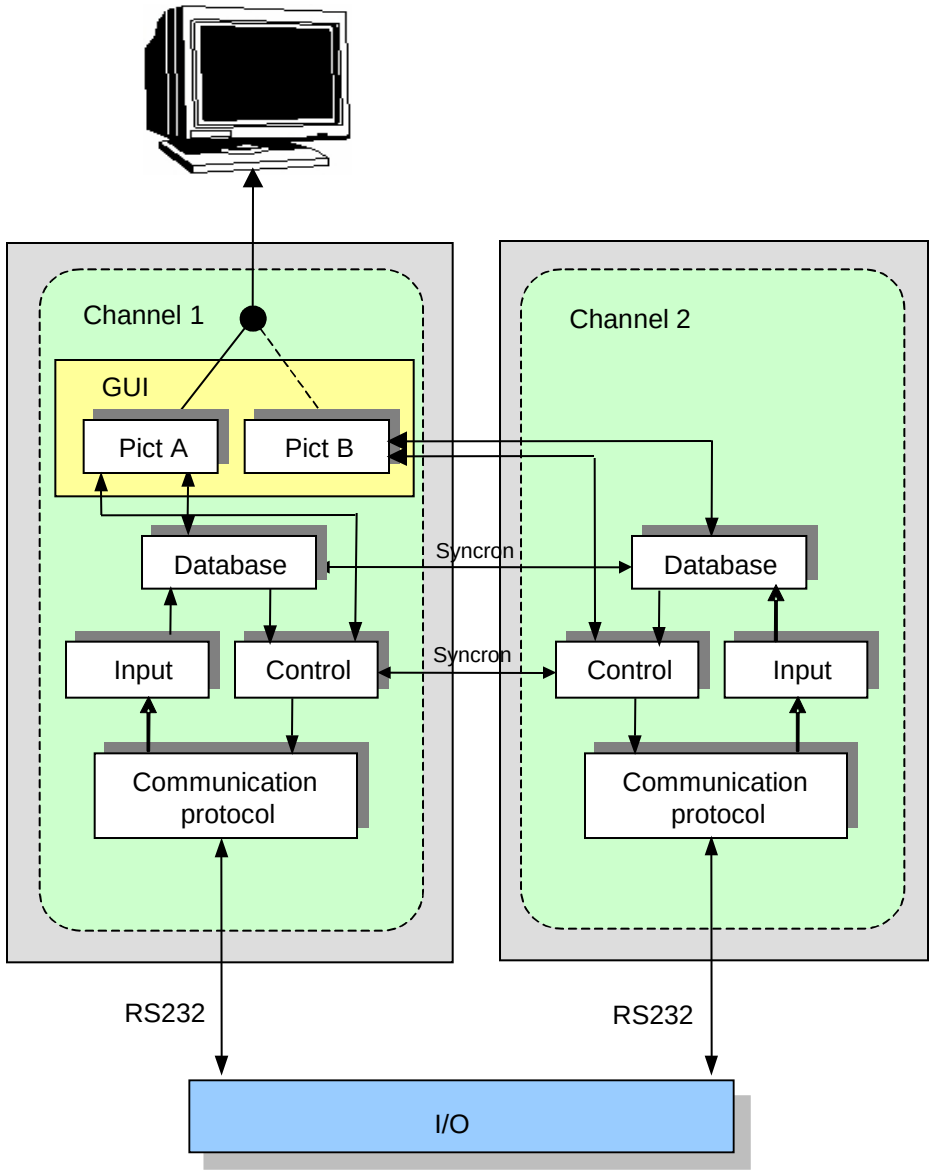
Példa: TI Hercules Safety Microcontrollers



Példa: SCADA rendszer

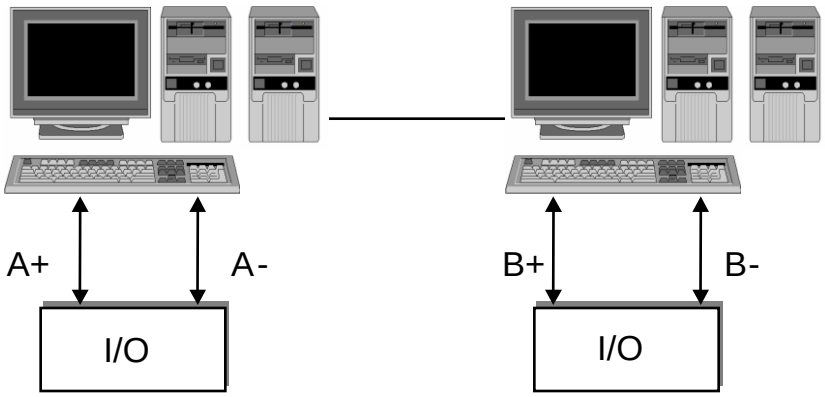


- Két csatorna
- Váltakozó bitmap megjelenítés (az operátor komparál)
- Szinkronizáció: Belső hibadetektálás (mielőtt a kimenetre kerülne)



Példa: SCADA telepítési opciók

- Két csatorna ugyanazon a szerveren
  - Statikusan linkelt szoftver modulok
  - Időben, memóriában és diszken elkülönülő végrehajtás
  - Diverz adattárolás
    - Bináris adatokra (jelek): Inverz adatábrázolás
    - Különböző adatbázis indexelés
- Két csatorna különböző szervereken
  - Szinkronizáció dedikált belső hálózaton
- Rendelkezésre állás növelése (hibatűrés):
  - Kétszer „2-ből 2” séma



Példa: SCADA hibadetektálás

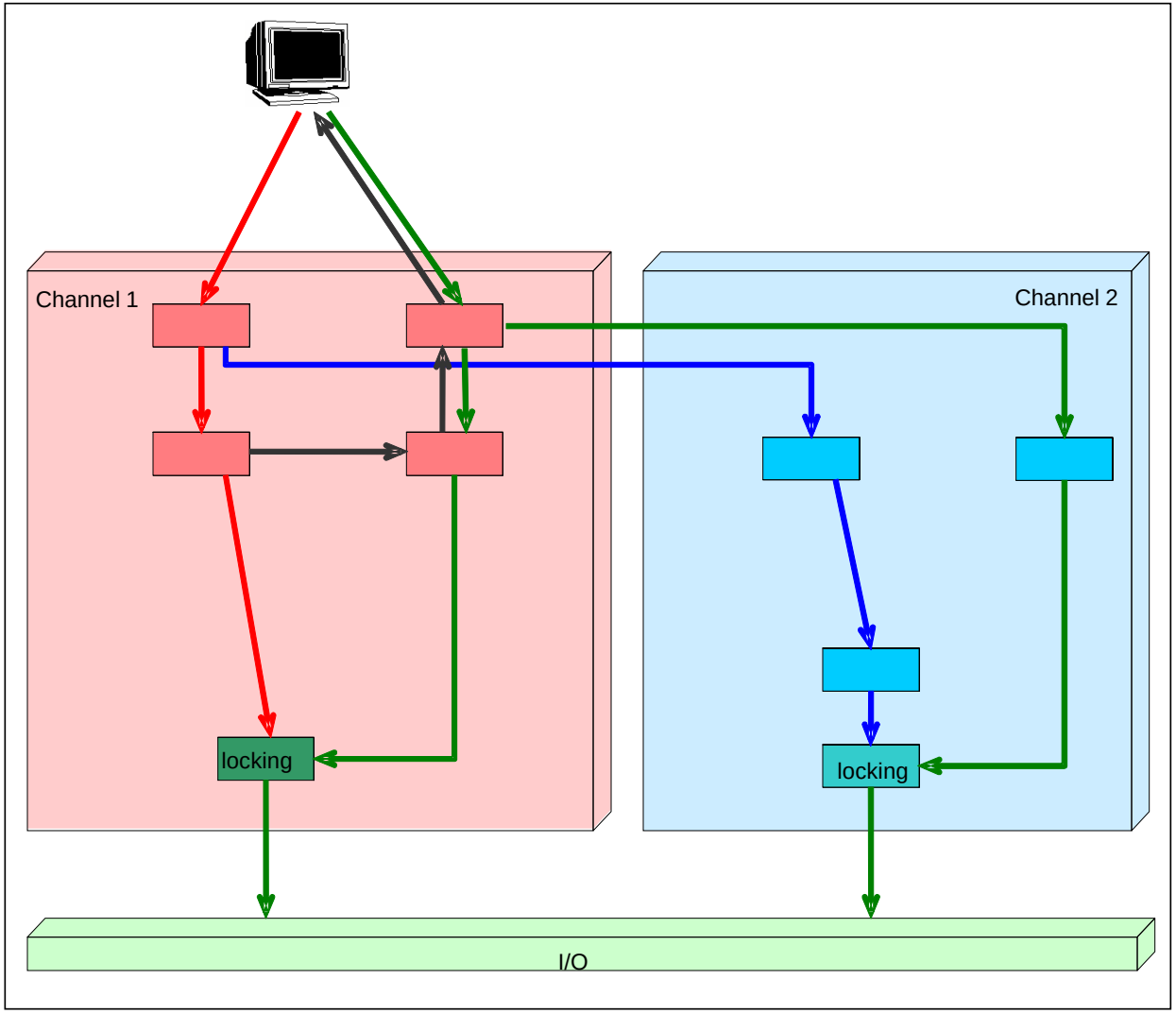
Véletlen hardver hibákra működés közben:

- Csatornák komparálása: Operátor illetve I/O
  - Operátornak: Villogó RGB-BGR szimbólum jelzi a frissítést
- Watchdog processz
  - Többi processz futásának ellenőrzése
- Az adatbázis tartalmának rendszeres összehasonlítása
  - Lappangó hibák detektálása

Szándékolatlan vezérlésre, szoftver hibákra:

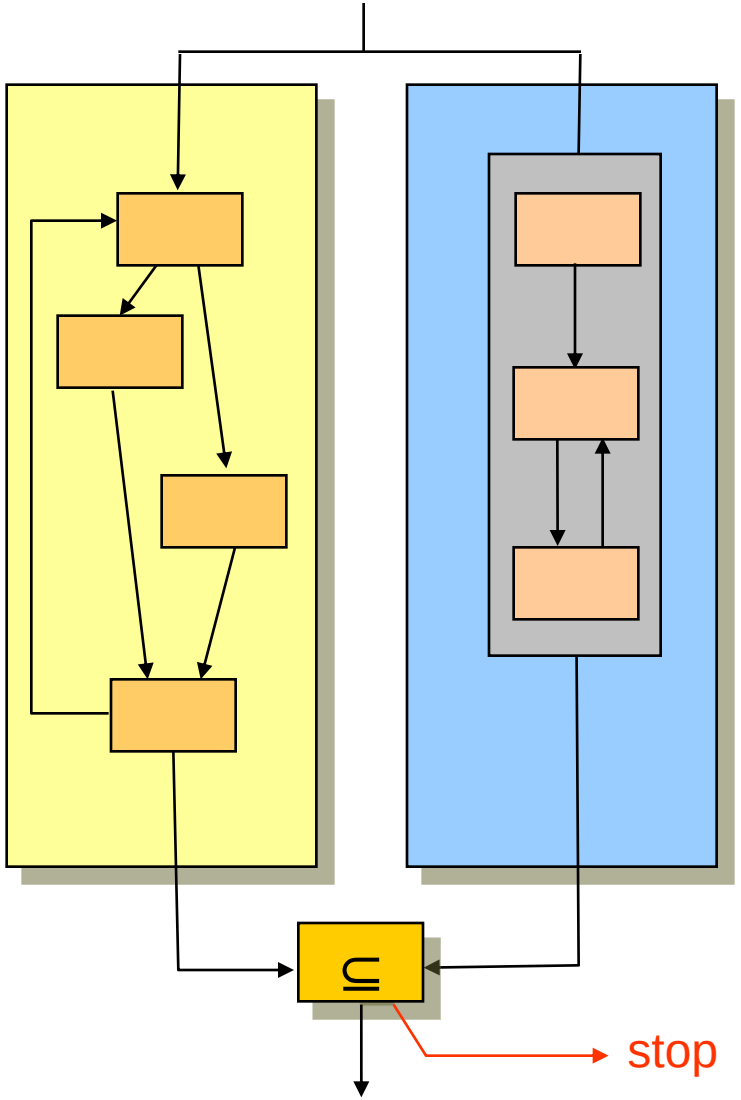
- Háromfázisú parancskiadás
  - Előkészítés (zárolva), visszaolvasás, független jóváhagyás
  - Diverz modulok a visszaolvasásra és jóváhagyásra

Példa: Háromfázisú parancskiadás

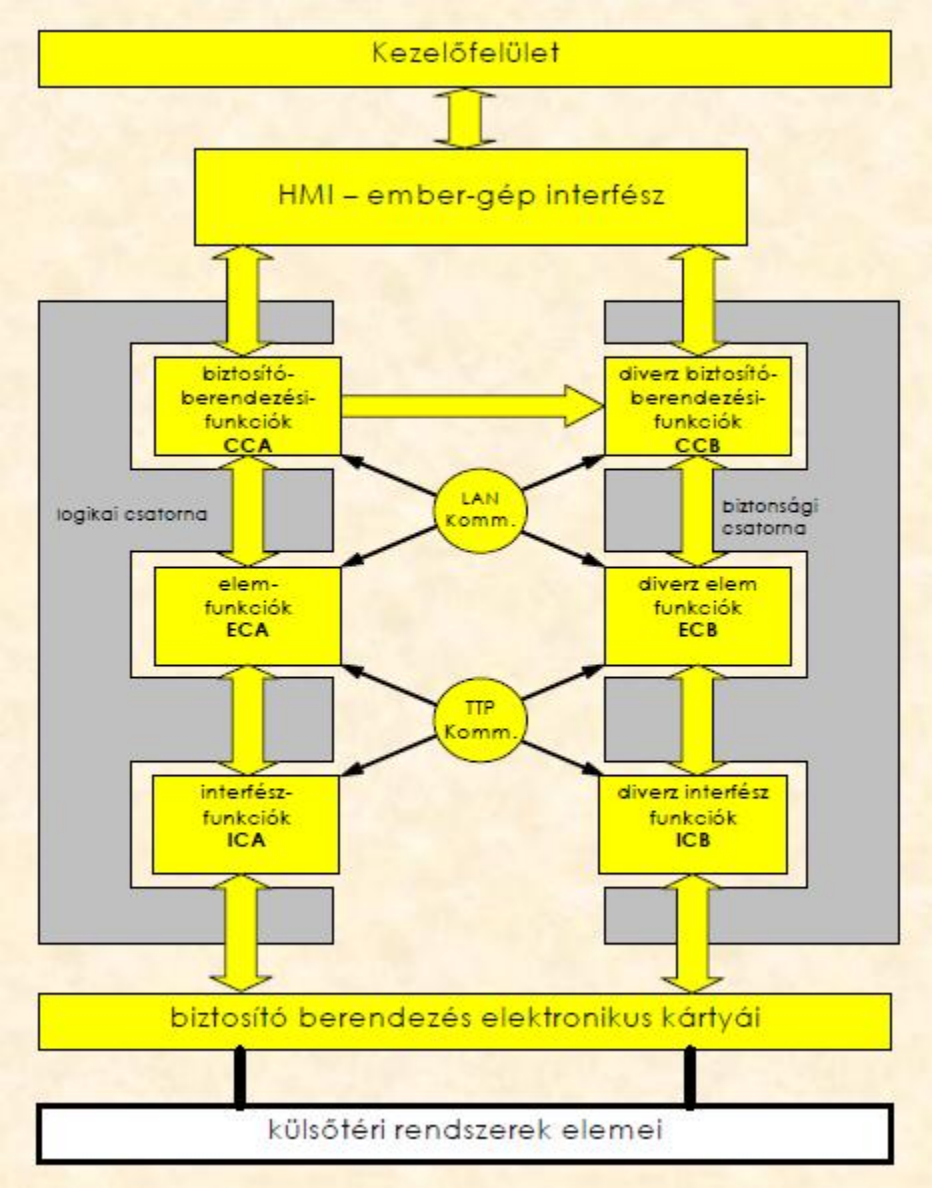


III. Kétcsatornás feldolgozás független ellenőrzéssel

- Független csatorna
  - „Safety bag”: csak biztonsági ellenőrzés
  - Eltérő tervezés
  - Megengedhető viselkedés ellenőrzése
- Példa:
  - Szabályok az elsődleges csatorna működésének ellenőrzésére



Példa: Alcatel (Thales) Elektra



Két csatorna:

- Logikai csatorna: CHILL (CCITT High Level Language) eljárás-orientált programnyelv
- Biztonsági csatorna: PAMELA (Pattern Matching Expert System Language) szabály-orientált programnyelv

Jellegzetes megoldások fail-operational rendszerekhez

- Hibadetektálás (és lekapcsolás) nem elegendő
- Tipikus megoldások
  - Időleges hibák:
    - Ismételt végrehajtás (újrapróbálás)
    - Helyreállítás
  - Állandósult működési hibák:
    - Hibadetektálás és tartalékra kapcsolás
    - Hibás komponens(ek) maszkolása:
      - Triple Modular Redundancy (TMR),
      - N-Modular Redundancy (NMR)
  - Állandósult tervezési hibák:
    - Eltérő megvalósítás (diverzitás)