

Hibatűrés

Majzik István

Budapesti Műszaki és Gazdaságtudományi Egyetem

Méréstechnika és Információs Rendszerek Tanszék

<http://www.mit.bme.hu/>

Hibatűrés különféle hibák esetén

- **Hardver tervezési hibák** (< 1%):
 - Tipikusan nem számítanak rá (jól tesztelt komponensek)
 - Eltérő tervezésű hardver lenne szükséges
- **Hardver állandósult működési hibák** (~10%):
 - Hardver redundancia (pl. tartalék processzor)
- **Hardver időleges működési hibák** (~70-80%):
 - Szoftver redundancia (pl. állapotmentés és helyreállítás)
 - Idő redundancia (pl. utasítás újravégrehajtás)
 - Információ redundancia (pl. hibajavító kódolás)
- **Szoftver tervezési hibák** (~20-30%):
 - Szoftver redundancia (pl. eltérő tervezésű modulok)

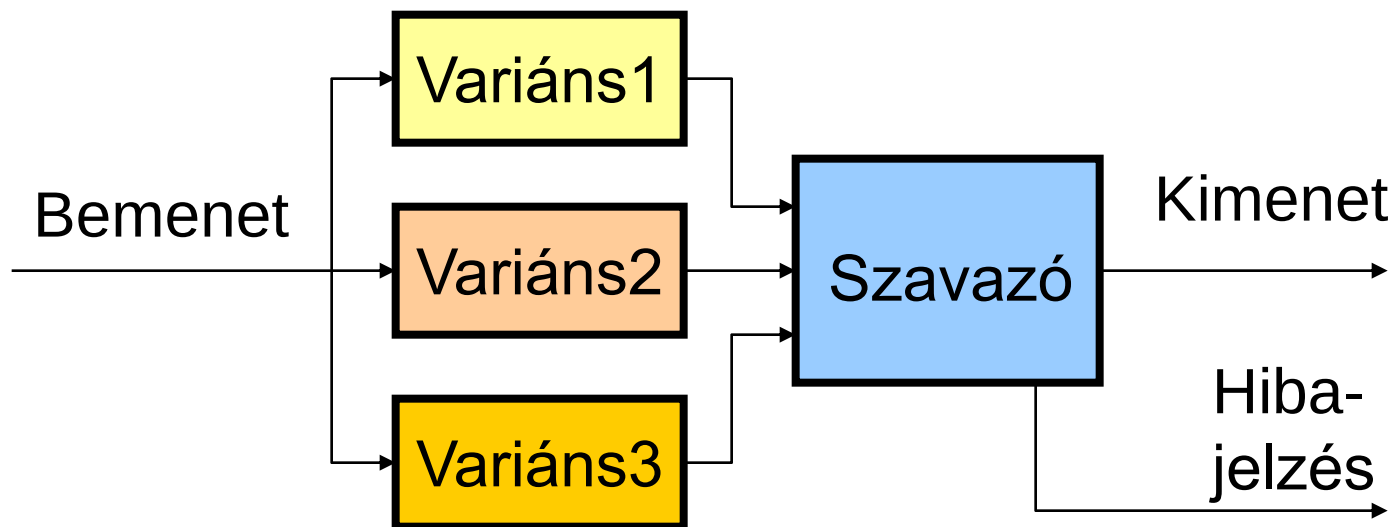
Hibatűrés szoftver tervezési hibák esetén

Szoftver tervezési hibák

- Ismételt végrehajtás nem segít (állandósult a hiba)
- Redundáns modulok: **Eltérő tervezés** szükséges
- **Variánsok**: azonos specifikáció, de
 - eltérő algoritmus és/vagy adatstruktúrák (diverzitás)
 - más fejlesztési környezet, programnyelv
 - elszigetelt fejlesztés**az azonos hibák bekövetkezésének csökkentésére**
- Variánsok végrehajtásának technikái:
 - N-verziós programozás (NVP: N-version programming)
 - Javító blokkok (RB: Recovery block)
 - N-önellenőrző programozás (NSCP: N-self-checking programming)
 - Önkonfiguráló optimista programozás (SCOP: Self-configuring optimistic programming)

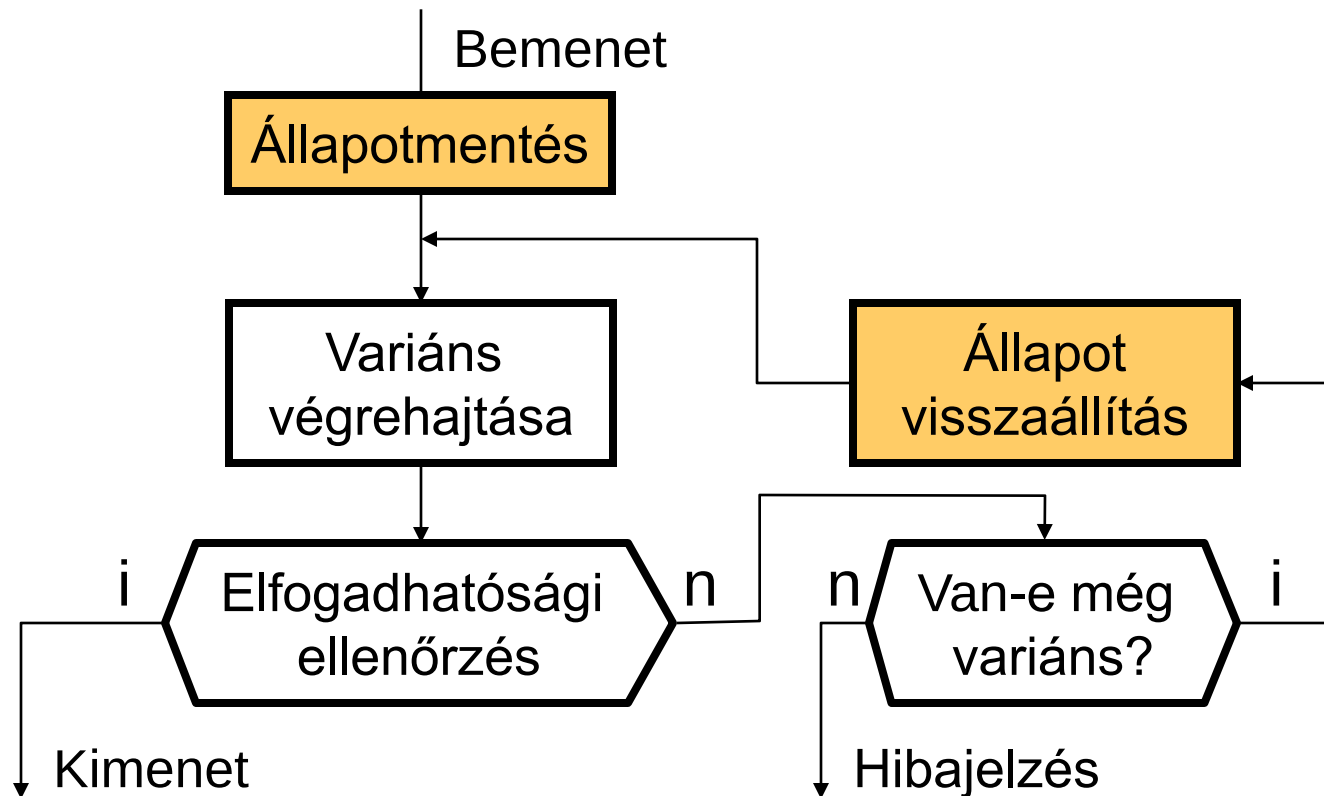
1. N-verziós programozás (NVP)

- **Aktív statikus** redundancia:
Minden variáns végrehajtása (tipikusan párhuzamosan)
 - Ugyanazon bemenetek
 - Többségi szavazás
 - Elfogadható értéktartományt kell megadni a kimenetekre
 - Szavazó egyszeres hibapont (SPOF), de egyszerű
- Kritikus komponens: Szavazó és a futtató környezet



2. Javító blokkok (RB)

- **Passzív** redundancia: Csak hiba esetén aktiválódik
 - Variánsok kimenetének **elfogadhatósági ellenőrzése**
 - Hiba esetén tartalék variáns (soros) végrehajtása

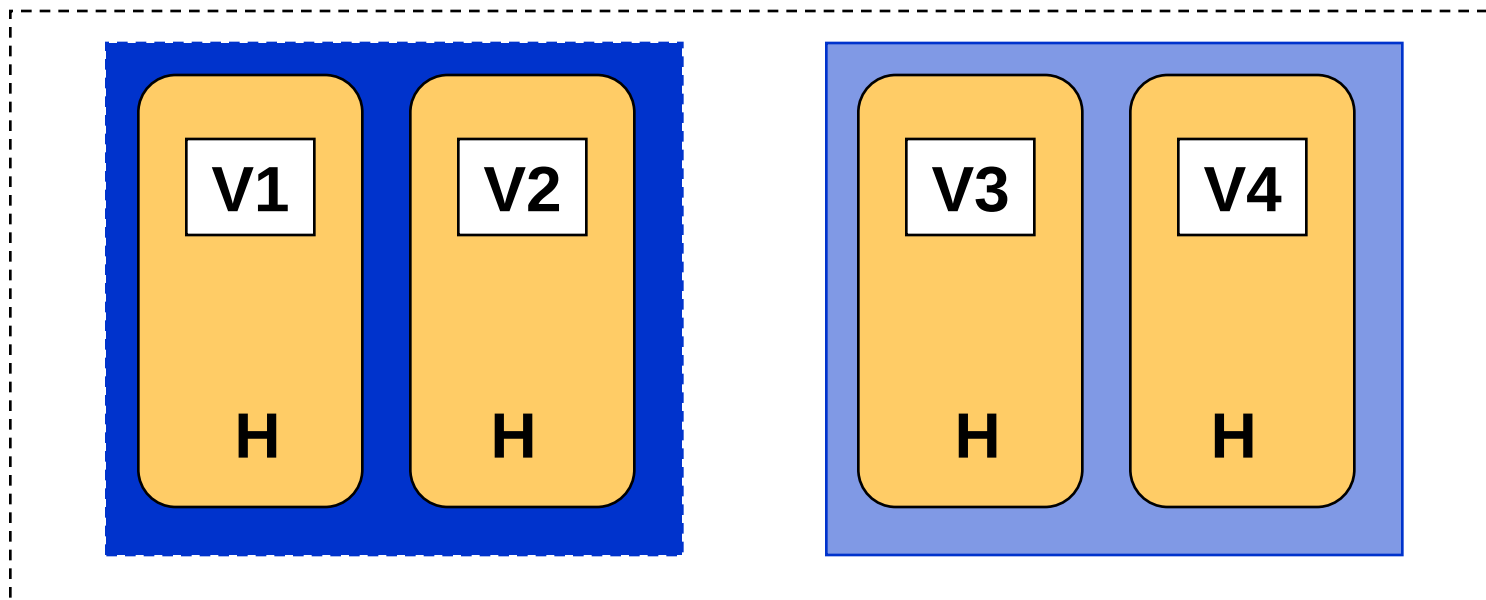


Összehasonlítás

Tulajdonság/típus	N-verziós prg.	Javító blokkok
Ellenőrzés	Szavazás, relatív	Elfogadhatóság abszolút
Végrehajtás	Párhuzamos	Soros
Időigény	Leglassabb variáns (vagy time-out)	Hibák számától függő
Redundancia	Mindig	Csak hiba esetén
Tolerált hibák	$[N/2-1]$	$N-1$
Hibakezelés	Maszkolás	Helyreállítás

3. N-önellenőrző programozás (NSCP)

- **Aktív dinamikus** redundancia:
N számú **önellenőrző** komponens
- Hibadetektálás esetén átkapcsolás az elsődlegesről a (következő) önellenőrző tartalékra
- Jellegzetes példa: Airbus A-320 fedélzeti számítógép
 - Páronként önellenőrző végrehajtás (szoftver variánsok)
 - Elsődleges pár működik, **átkapcsolás hiba esetén**
 - Állandósult hardver hiba: Ismételten hibázó pár **lekapcsol**



4. Önkonfiguráló optimista programozás (SCOP)

Adaptív redundancia: Csak a szükséges számú variáns

- Paraméterek:
 - Hány variáns egyetértése (azonos kimenet) szükséges: a
 - Hány variáns van: n
 - Mennyi idő áll rendelkezésre: t
- Iteratív végrehajtás
 1. Kezdetben a minimális számú (várhatóan egyetértő) variáns végrehajtása: a számú variáns
 2. Kimenetek ellenőrzése komparálással: k egyezik
 - Ha $k=a$, akkor kilépés: OK
 - Ha $k < a$, akkor (optimista módon) $a-k$ újabb variáns végrehajtása az egyetértés eléréséhez, és folytatás a 2. lépéssel
 - Kilépés hibajelzéssel:
Ha nincs több végrehajtható variáns, vagy letelt a t idő

Adat diverzitás

- Alapötlet:
 - Azonos feldolgozási algoritmus, de **eltérő adatszerkezetű** variánsok (implementáció ezért némileg eltérő)
 - Pl. vektor vagy láncolt lista
 - Pl. más adatbázis kezelő, más lekérdezési mód
 - Másolatok (replikák): Azonos szoftver komponensek, de **eltérő állapotok** lehetnek
 - Pl. numerikus eltérések, más adatbázis példányok
- Végrehajtási módok:
 - **Újrapróbáló blokk** (retry block):
 - Javító blokkok analógiája: Passzív redundancia szerinti végrehajtás
 - **N másolat** (N-copy programming):
 - NVP analógiája: Aktív redundancia szerinti (párhuzamos) végrehajtás

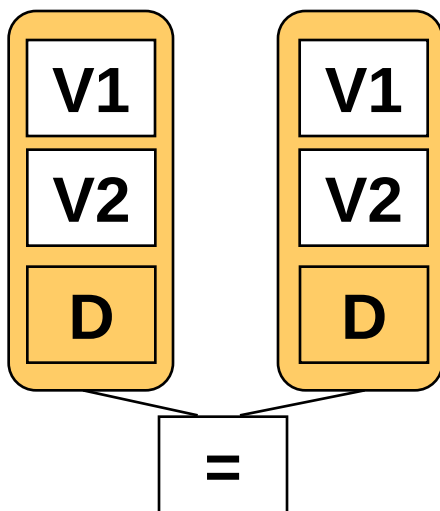
Hibrid architektúrák (áttekintés)

Típus / tolerált **hardver** állandósult hibák / tolerált **szoftver** tervezési hibák (maximális) száma

- RB/1/1
 - Elfogadhatósági teszt után **komparálás**
 - Diagnosztikai ellenőrzés után a hibás lekapcsol
- RB/2/1
 - Elfogadhatósági teszt után **szavazás**
 - Ismételten eltérő lekapcsol (RB/1/1 lesz)
- NVP/1/1
 - Variánsok közötti **szavazás**
 - Ismételten eltérő lekapcsol (komparálás marad)
- NVP/2/1
 - 4 variáns esetén **szavazás**
 - Ismételten eltérő lekapcsol, újrakonfigurálás után NVP/1/1 lesz

Hibrid architektúrák 1.

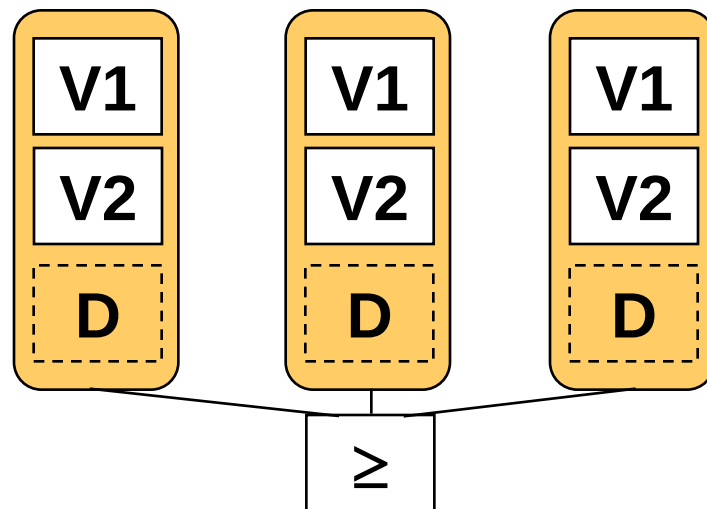
RB/1/1



RB/1/1

- Elfogadhatósági ellenőrzés után **komparálás**
- Diagnosztikai ellenőrzés után a hibás hardver lekapcsol

RB/2/1

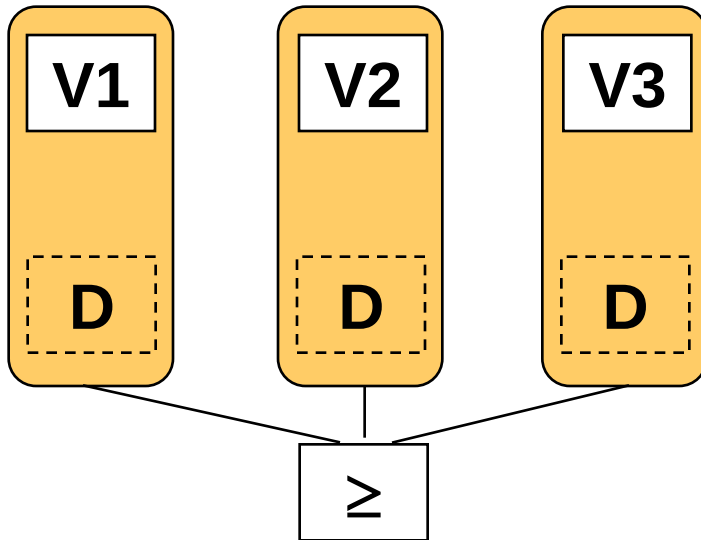


RB/2/1

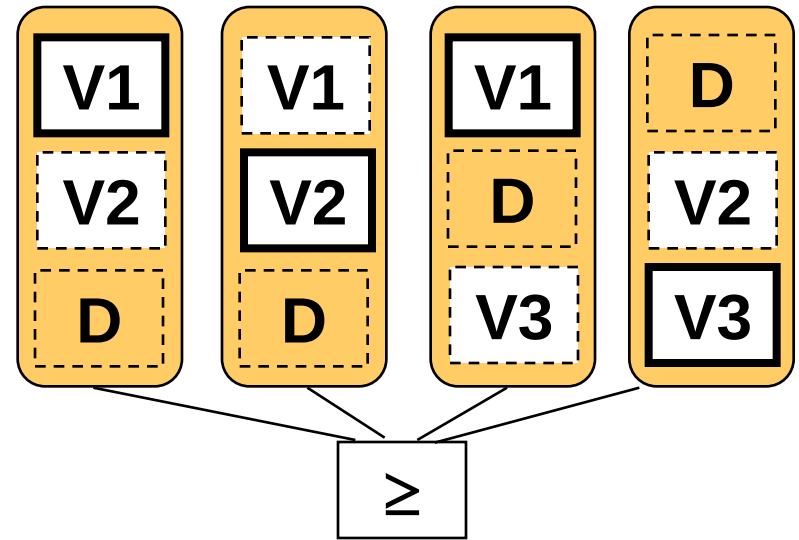
- Elfogadhatósági ellenőrzés után **szavazás**
- Ismételten eltérő lekapcsol (RB/1/1 marad)

Hibrid architektúrák 2.

NVP/1/1



NVP/2/1



NVP/1/1

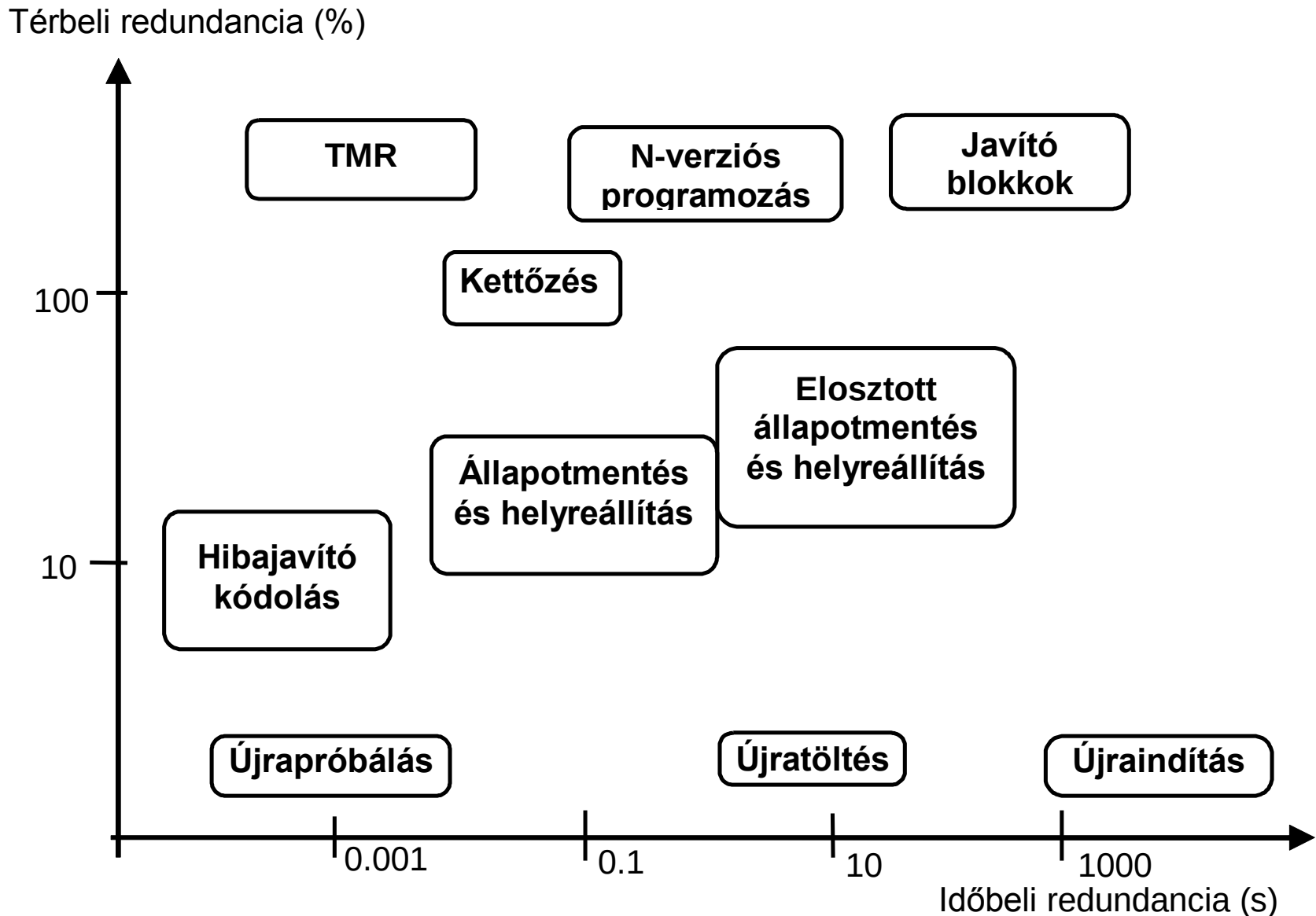
- Variánsok közötti **szavazás**
- Ismételten eltérő lekapcsol (komparálás marad, diagnosztikai ellenőrzéssel)

NVP/2/1

- 4 variáns esetén **szavazás**
- Ismételten eltérő lekapcsol (NVP/1/1 konfigurálható)

Összefoglalás

Hibatűrési technikák összehasonlítása



Szabvány szerinti módszerek

- IEC 61508:
Functional
safety in
electrical /
electronic /
programmable
electronic
safety-related
systems
- Szoftver
architektúra
tervezés

**Table A.2 – Software design and development:
software architecture design (see 7.4.3)**

Technique/Measure*		Ref	SIL1	SIL2	SIL3	SIL4
1	Fault detection and diagnosis	C.3.1	---	R	HR	HR
2	Error detecting and correcting codes	C.3.2	R	R	R	HR
3a	Failure assertion programming	C.3.3	R	R	R	HR
3b	Safety bag techniques	C.3.4	---	R	R	R
3c	Diverse programming	C.3.5	R	R	R	HR
3d	Recovery block	C.3.6	R	R	R	R
3e	Backward recovery	C.3.7	R	R	R	R
3f	Forward recovery	C.3.8	R	R	R	R
3g	Re-try fault recovery mechanisms	C.3.9	R	R	R	HR
3h	Memorising executed cases	C.3.10	---	R	R	HR
4	Graceful degradation	C.3.11	R	R	HR	HR
5	Artificial intelligence - fault correction	C.3.12	---	NR	NR	NR
6	Dynamic reconfiguration	C.3.13	---	NR	NR	NR
7a	Structured methods including for example, JSD, MASCOT, SADT and Yourdon.	C.2.1	HR	HR	HR	HR
7b	Semi-formal methods	Table B.7	R	R	HR	HR
7c	Formal methods including for example, CCS, CSP, HOL, LOTOS, OBJ, temporal logic, VDM and Z	C.2.4	---	R	R	HR
8	Computer-aided specification tools	B.2.4	R	R	HR	HR

NOTE – The measures in this table concerning fault tolerance (control of failures) should be considered with the requirements for architecture and control of failures for the hardware of the programmable electronics in IEC 61508-2.

* Appropriate techniques/measures shall be selected according to the safety integrity level. Alternate or equivalent techniques/measures are indicated by a letter following the number. Only one of the alternate or equivalent techniques/measures has to be satisfied.

Hibatűrés verifikációja

Meghibásodás (hiba) előidézése: **Hibainjektálás**

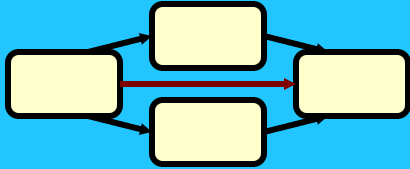
- Modell alapon (tervezés közben):
 - Ellenőrzés szimulációval, modellellenőrzéssel
- Prototípus esetén (tesztelés, robusztusság tesztelés):
 - **Hardver hibainjektálás:**
 - Meghibásodás (hibaok) létrehozása:
pl. busz jelek befolyásolása, tápfeszültség túske előidézése, nehézion-besugárzás, hőterhelés
 - Hardverfüggő, lassú, drága
 - **Szoftver hibainjektálás:**
 - Hibaállapot létrehozása (rendszerállapot megváltoztatása):
pl. regiszterek, memóriabitek átállítása
 - Szoftver tervezési hiba bevitele kód mutációval
 - Rugalmas, gyors, olcsóbb
 - Hibaoknak való megfelelés kérdéses
 - **Hibrid hibainjektálás**
 - Szoftver és hardver megoldások ötvözése

Hibainjektáló eszközök (példák)

- **Modell alapú** injektálás:
 - VHDL, HDL szint: FOCUS, MEFISTO
 - Komponens szint (CPU, diszk, memória): DEPEND
- **Hardver** hibainjektálás:
 - Közvetlen hozzáférés jelekhez: RIFLE, GOOFI
 - Besugárzás (nehéz-ion, neutron), hőkamra
- **Szoftver** hibainjektálás:
 - Hardver hibaok hatásának szoftver emulációja: FIAT, FERRARI, FTAPE
 - Alacsony szintű emuláció: DOCTOR, Xception
 - Kód mutációs eszközök: FINE, DEFINE, G-SWFIT
 - Protokoll rétegek hibái: ORCHESTRA, Neko, WS-FIT

A hibainjektálás absztrakciós szintjei (hardver)

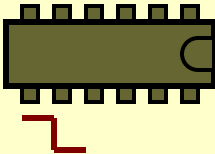
Ár,
valóságűség



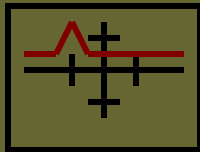
Hibaszimuláció a forráskódban
vagy modell szinten a fejlesztőeszközben



Regisztartalom módosítása,
memóriakép felülírása



Síneken haladó vagy áramkörök
lábán megjelenő jelek módosítása



Radioaktív sugárzás, ioninjektálás,
tápfeszültség zavarása, hőmérséklet

Összefoglalás: Hibatűrés technikái

1. Állandósult hardver hibák

- Hardver redundancia: többszörözés

2. Időleges hardver hibák

- Információ redundancia: PI. hibajavító kódolás
- Idő redundancia: PI. utasítás ismétlés
- Szoftver alapú hibatűrés
 1. Hibadetektálás
 2. Hibahatás felmérés
 3. Helyreállítás: visszalépő, előrelépő, kompenzáció
 4. Hibaok kezelése

3. Szoftver tervezési hibák

- Eltérő tervezésű variánsok futtatása: NVP, RB, NSCP, SCOP