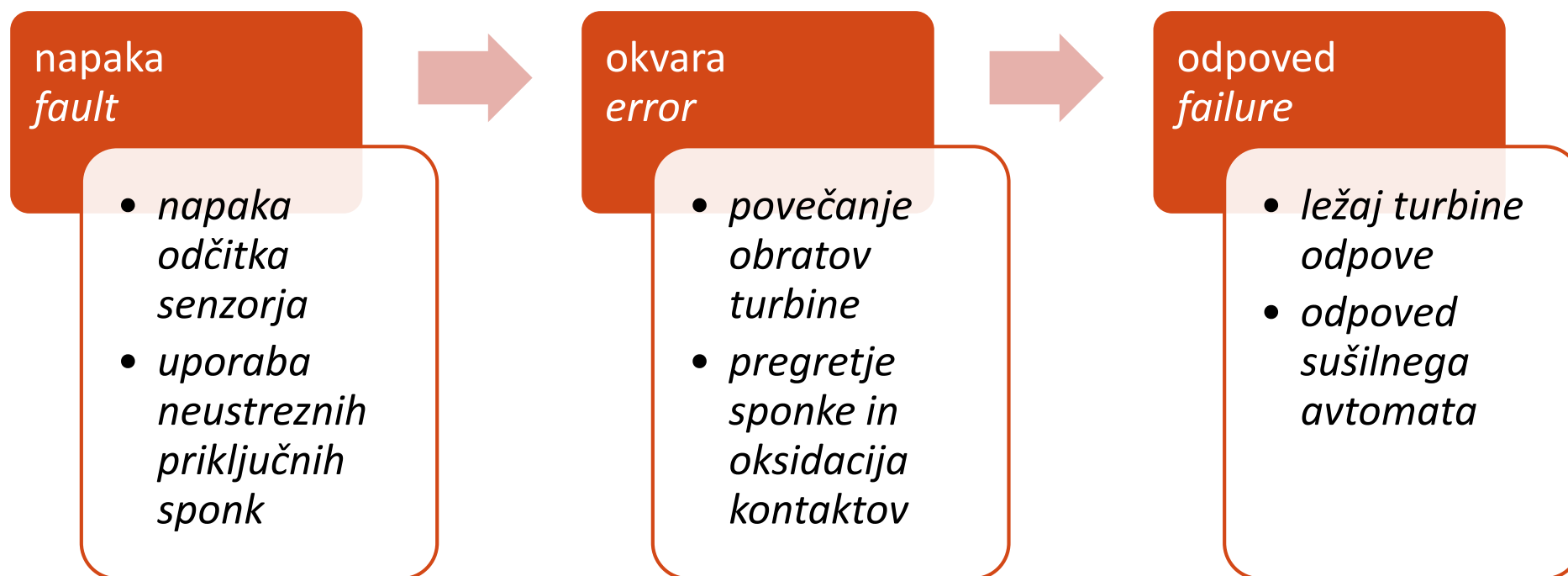


REALIZACIJA ELEKTRONSKIH SKLOPOV

Detekcija in odprava napak

M. Jankovec

Evolucija napake



Viri napak

Nepravilne
specifikacije

Nepravilne
implementacije

Defekti
komponent

Vplivi okolja

Tipi napak

Glede na izvor

- Strojne
- Programske

Glede na trajanje

- Tranzientne
- Ponavljajoče
- Permanentne

Odpornost na napake

Maskiranje

Detekcija



Lociranje

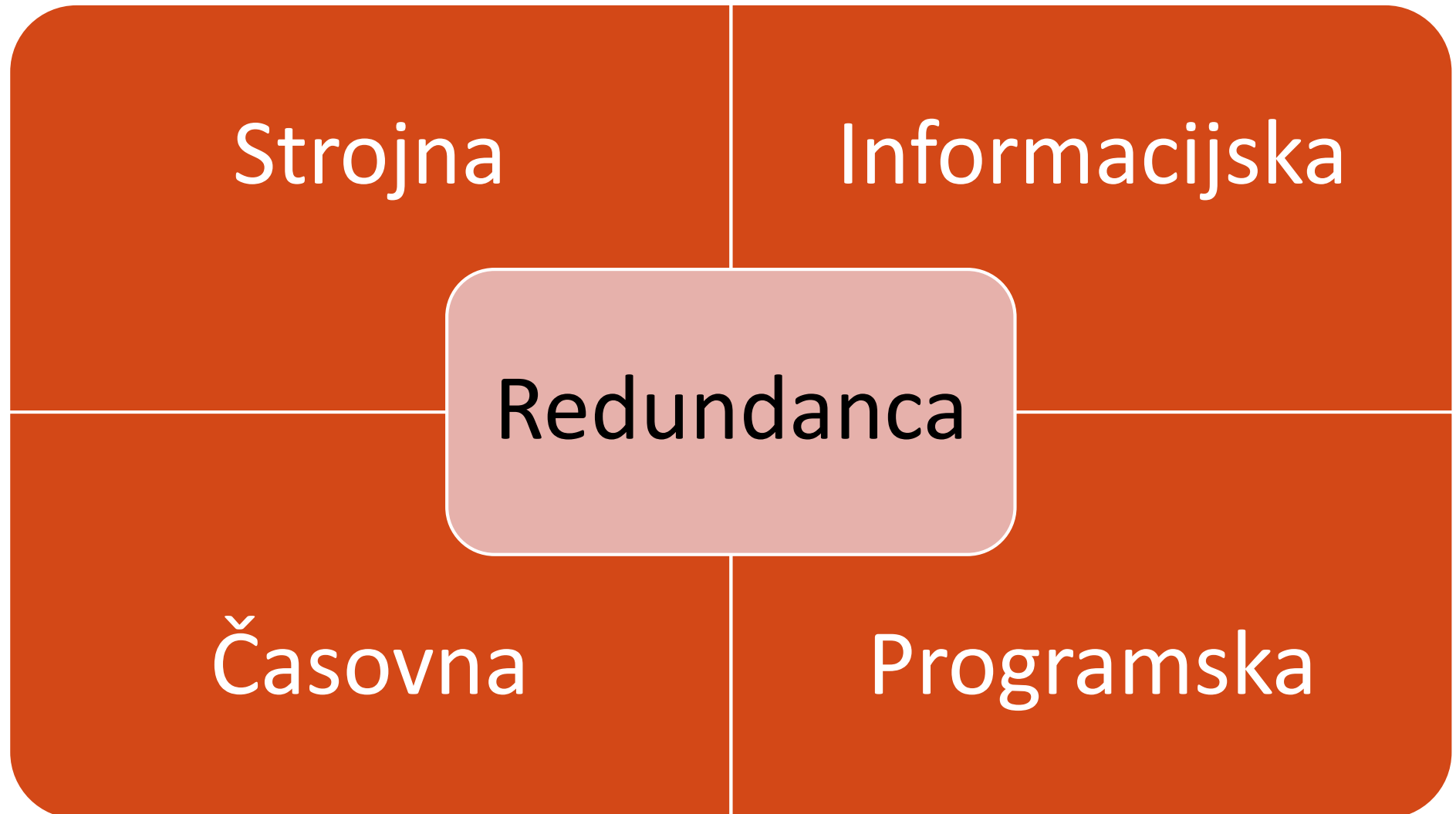


Izolacija



Poprava

Povečanje zanesljivosti sistemov



Strojna redundanca

Pasivna

- maskiranje napake

Aktivna

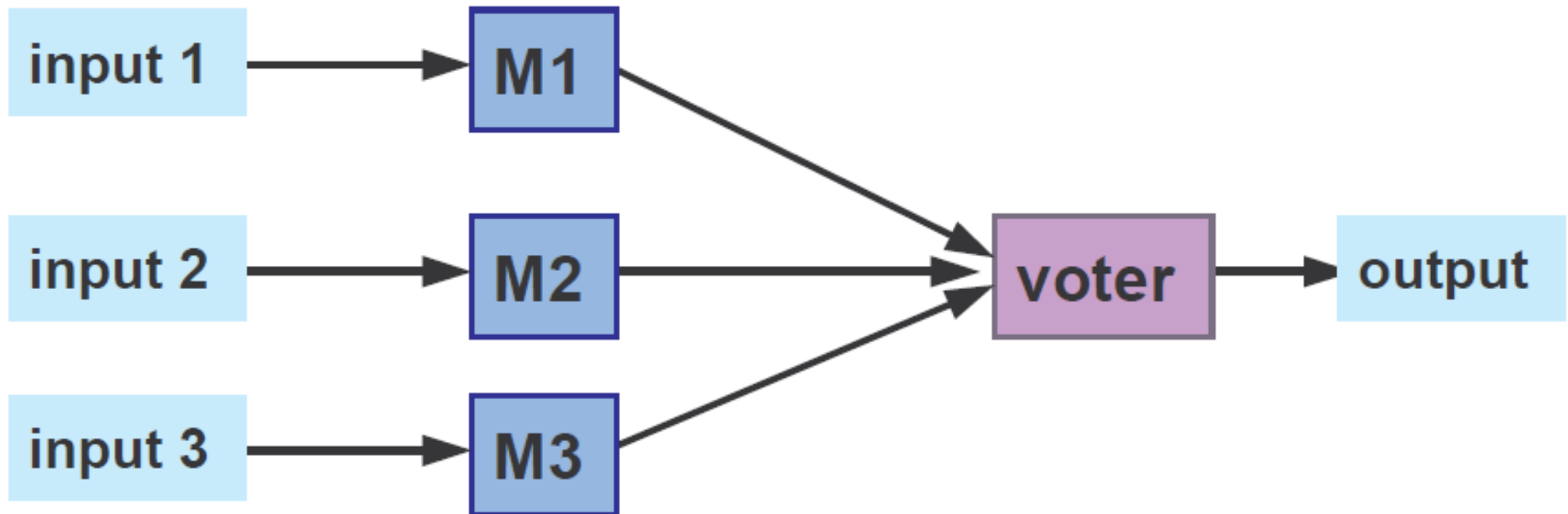
- detekcija
- lociranje
- omejitev
- odprava

Hibridna

- Kombinacija obeh

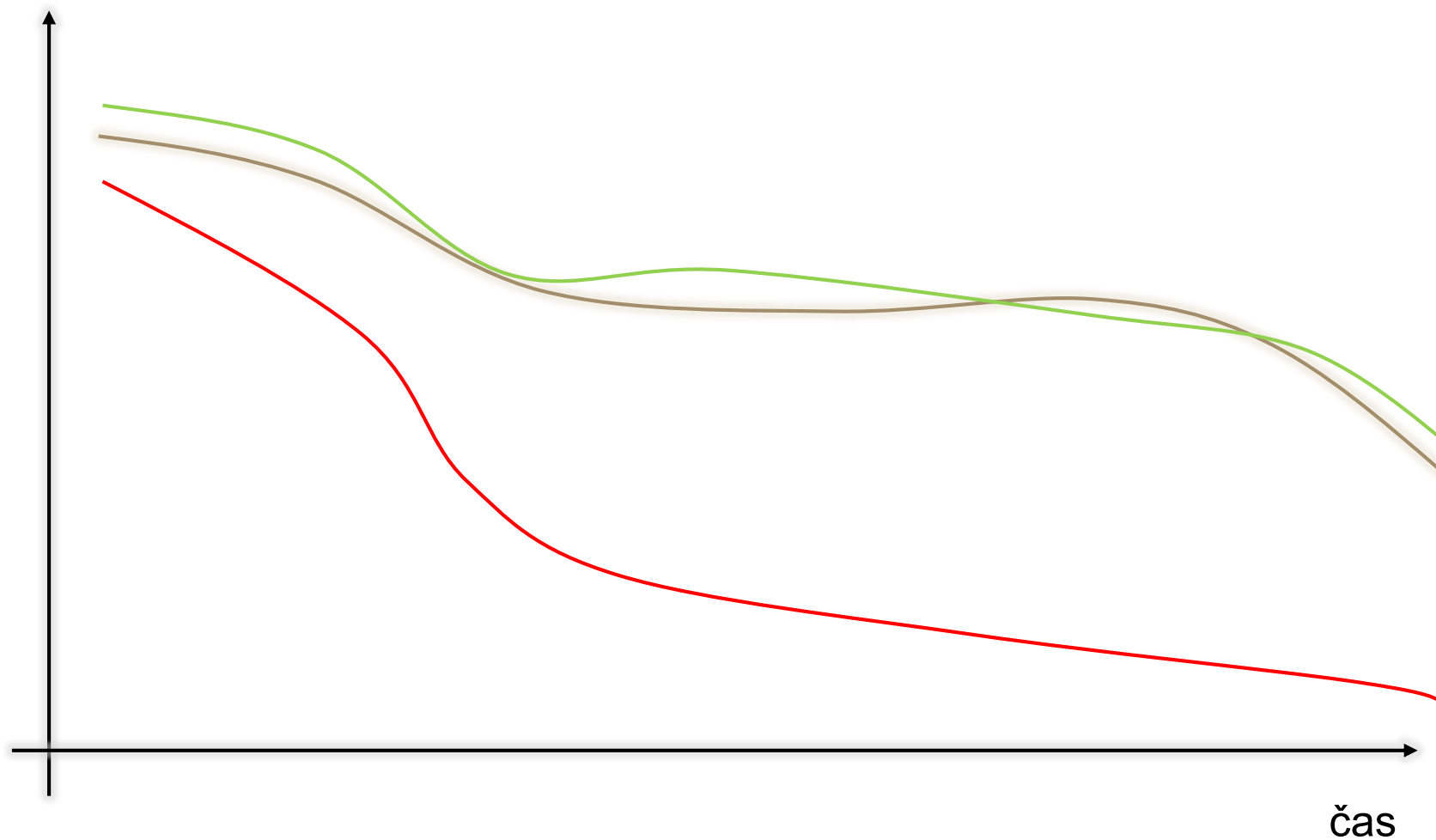
Pasivna redundanca

- Trojna modularna redundanca (TMR)
 - maskira lahko največ eno napako



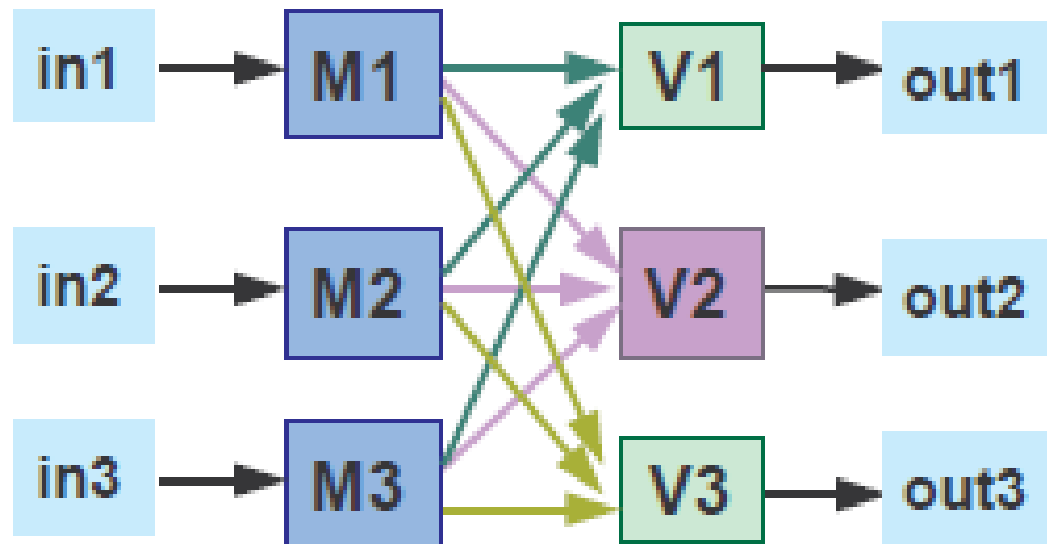
Odločitveni sklop

vrednost



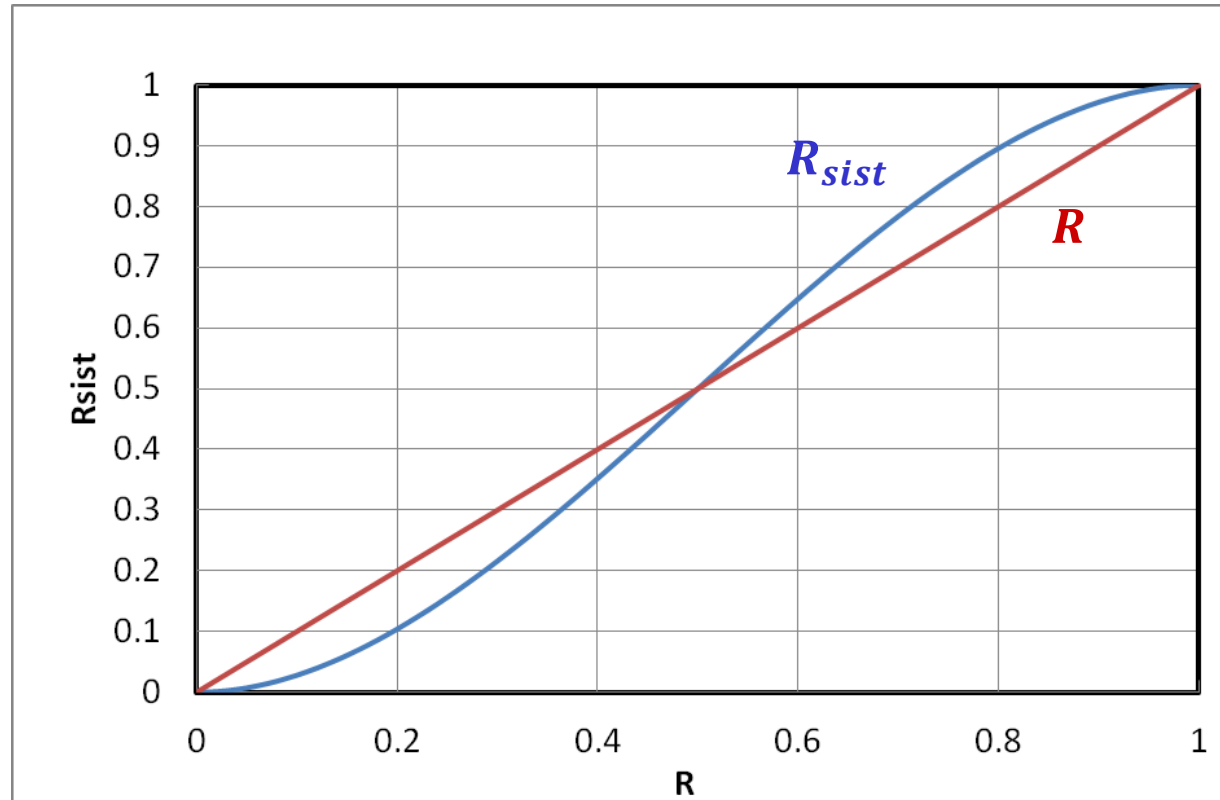
Potrojitev odločitvenega sklopa

- Napaka se lahko zgodi tudi v odločitvenem sklopu
 - $R_{TMR} = (3R^2 - 2R^3)R_v$
- Rešitev: redundanca odločitvenega sklopa



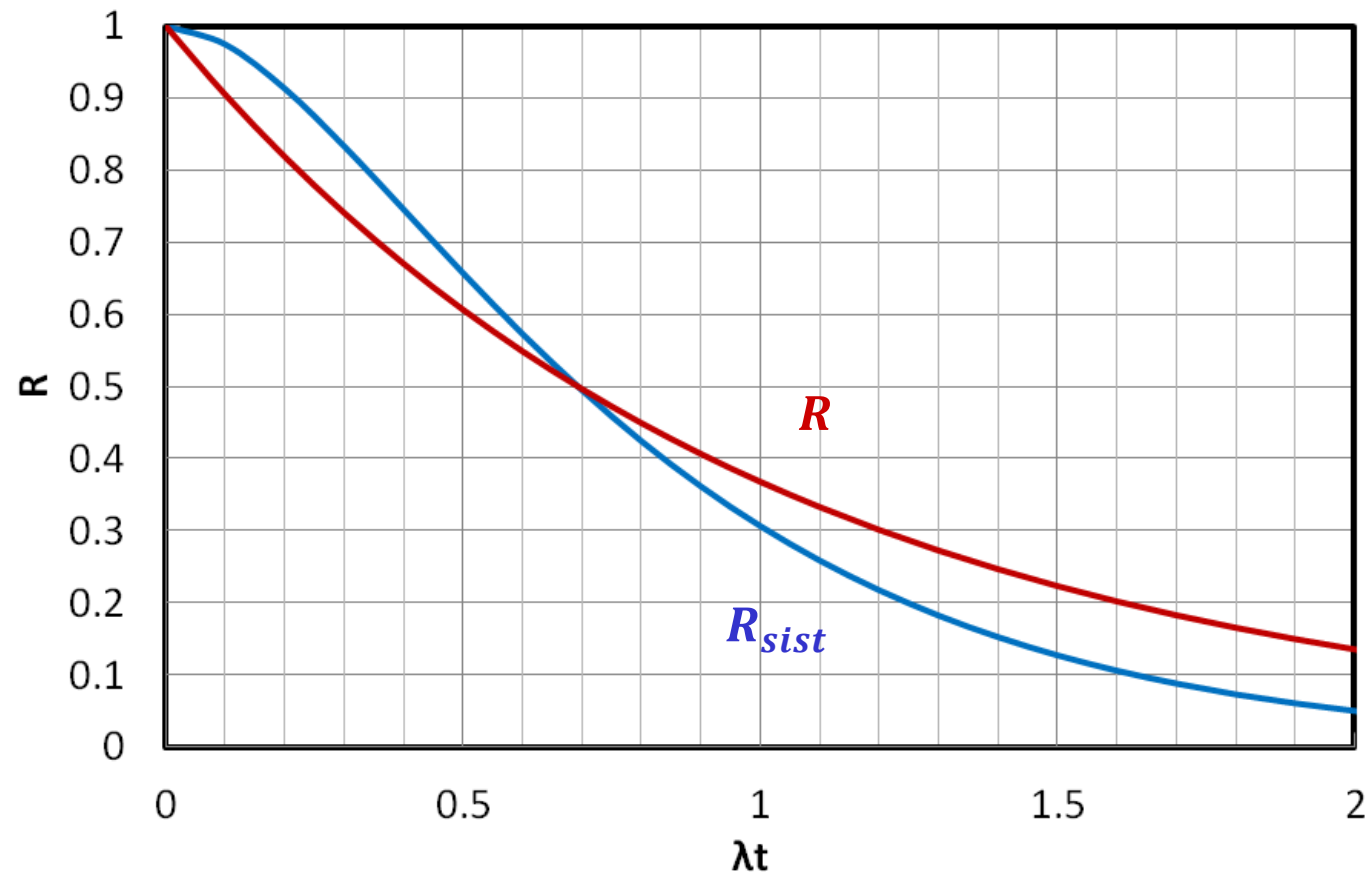
Zanesljivost TMR

- $R_{TMR} = R_1 R_2 R_3 + (1 - R_1) R_2 R_3 + R_1 (1 - R_2) R_3 + R_1 R_2 (1 - R_3)$
- Če velja $R_1 = R_2 = R_3$ sledi
- $R_{TMR} = 3R^2 - 2R^3$

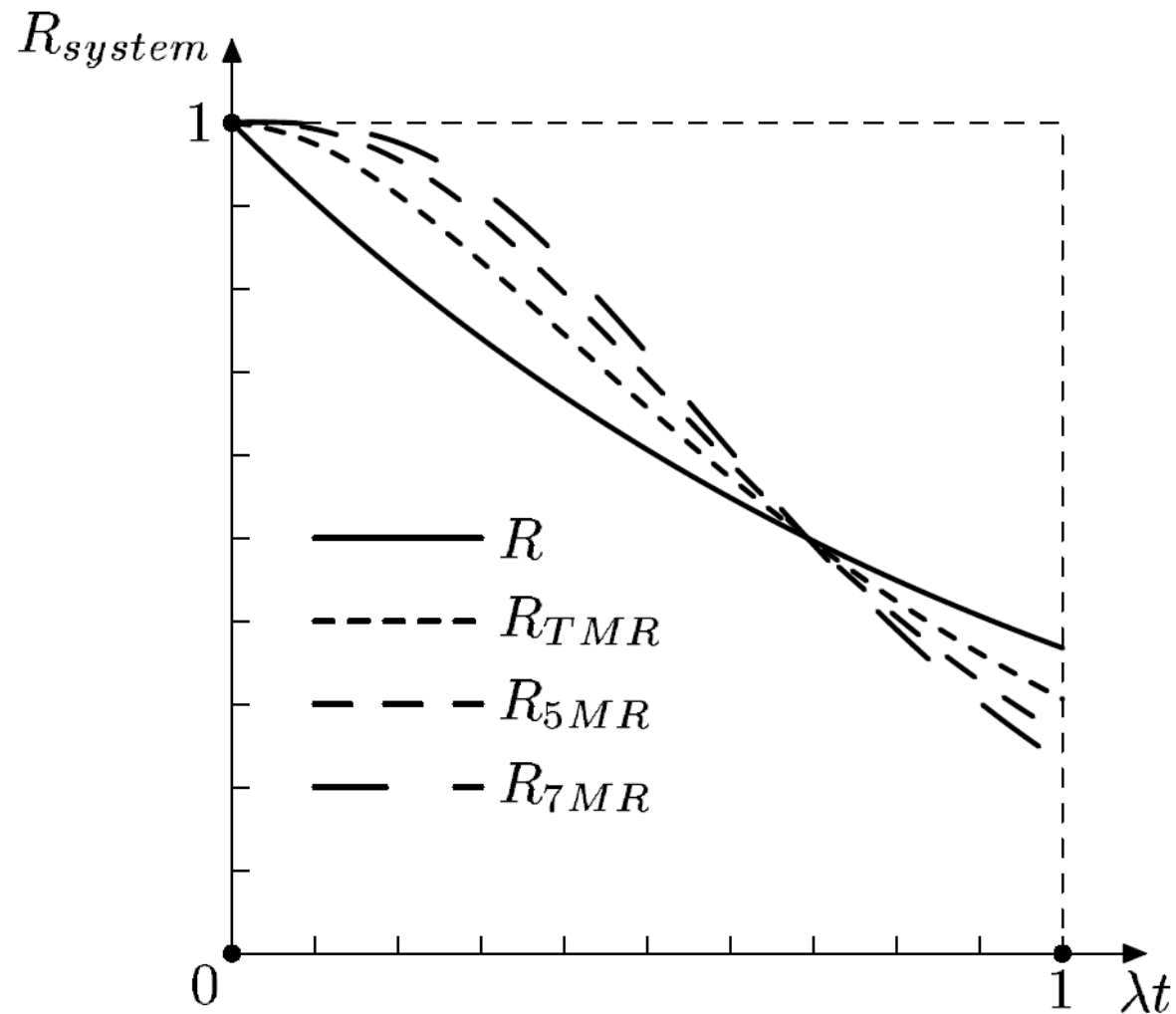


Časovni potek zanesljivosti TMR

- Če je $R(t) = e^{-\lambda t}$, potem je
- $R_{sist}(t) = 3e^{-2\lambda t} - 2e^{-3\lambda t}$



Več-kratne modularne redundance

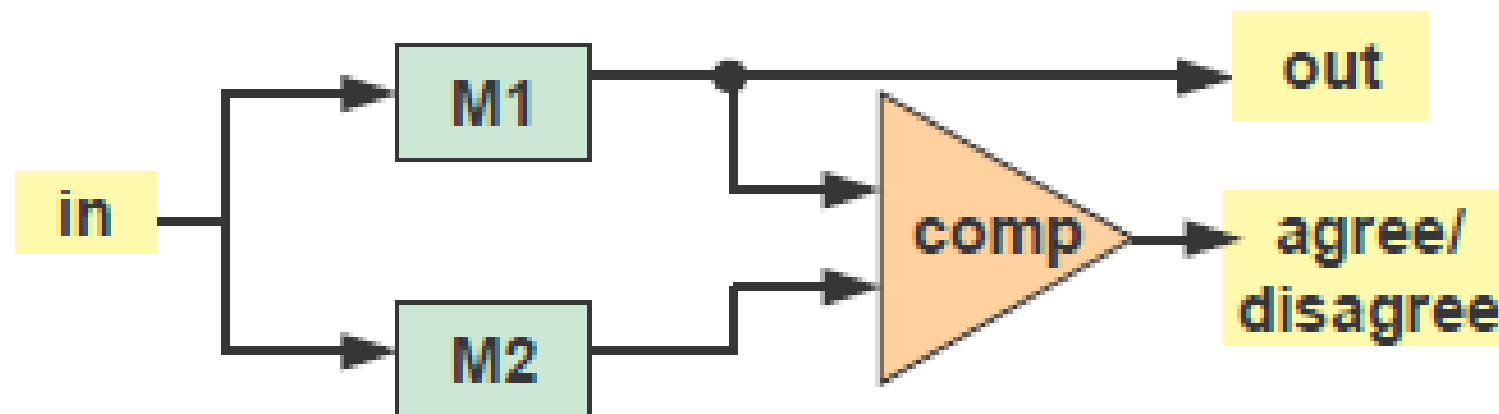


Aktivna redundanca

- Podvojitev s primerjavo
- Rezervni sistemi
 - Vroči
 - Hladni
- Nadzorni časovnik (Watchdog timer)

Podvojitev s primerjavo

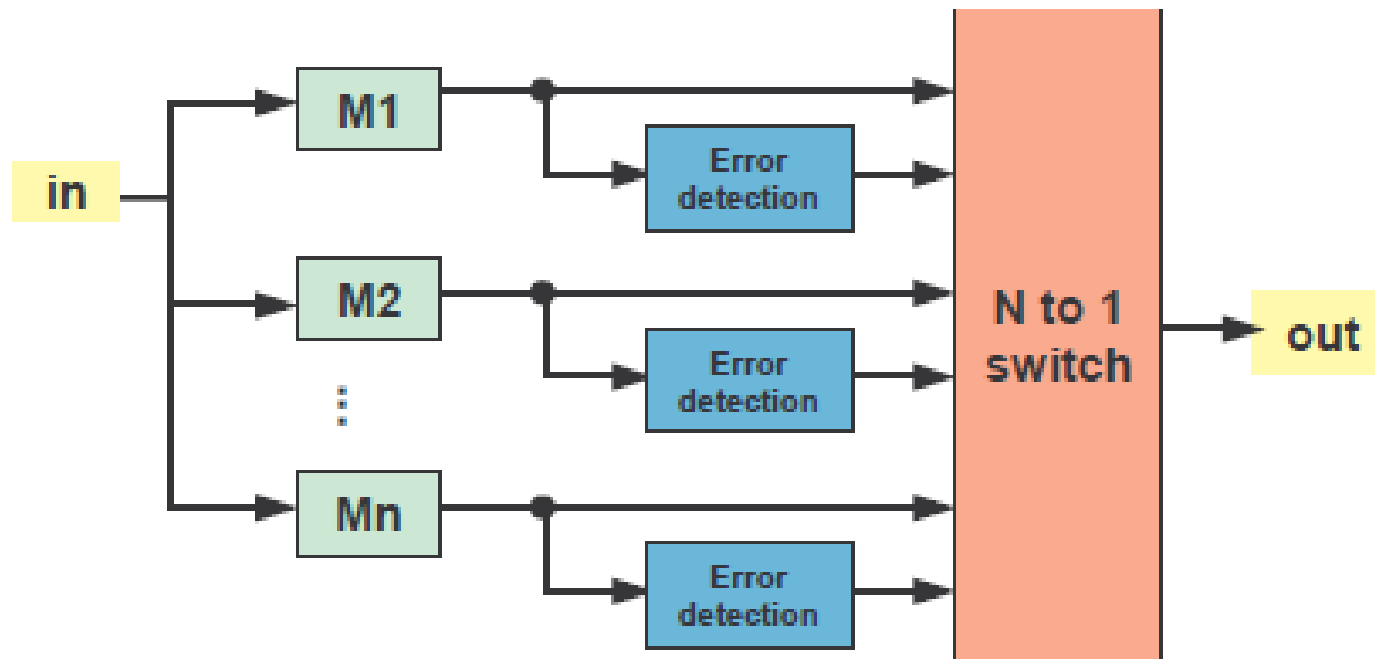
- Dva identična modula izvajata enako nalogo in njuna rezultata se primerjata



- Lahko detekira napako, ne more je pa popraviti
- Sistem deluje, če delujeta oba modula
 - $R_{DC} = R_1 R_2$
- Zanesljivost je manjša od posameznega modula

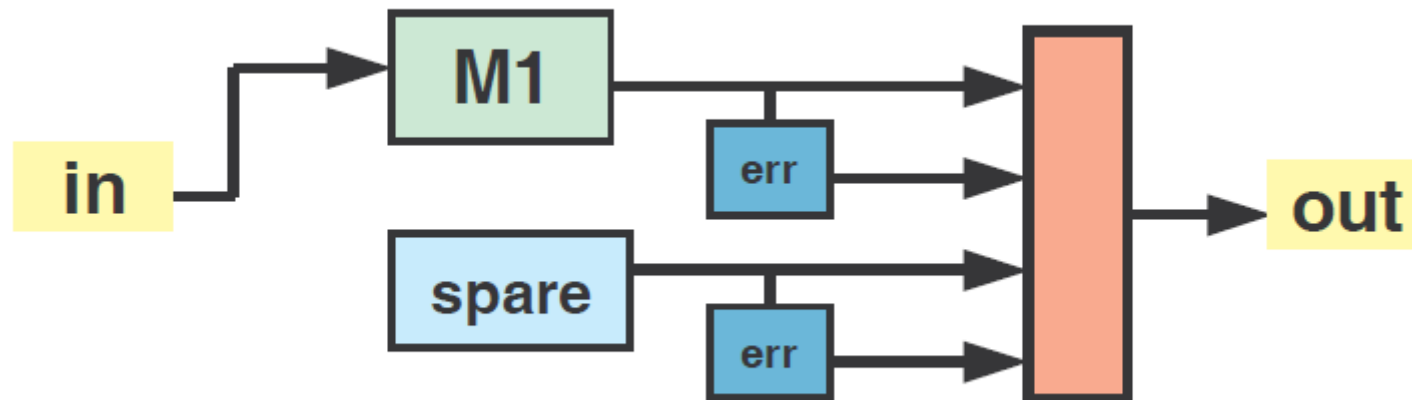
Rezervni sistemi

- Eden modul deluje, ostali so rezervni
- Izbira delujočega modula po prioritetni listi



Hladna rezerva

- Rezervni sistem, ki je med delovanjem primarnega sistema izklopljen.



- Ob izpadu primarnega sistema se vklopi rezervni sistem, ampak zaradi zagonskega časa lahko nastane motnja.

Zanesljivost sistema s hladno rezervo

- Sistem, ki lahko odpravi eno napako, deluje zanesljivo, če ni okvare (P_0) ali če je samo ena okvara (P_1).

$$R_{sist}(t) = P_0(t) + P_1(t) = e^{-\lambda t} + \frac{\lambda t}{1!} e^{-\lambda t}$$

- Če skušamo iz enačbe za zanesljivost izračunati življenjsko dobo (LT je čas t da dosežemo R_{min}), dobimo transendenčno enačbo

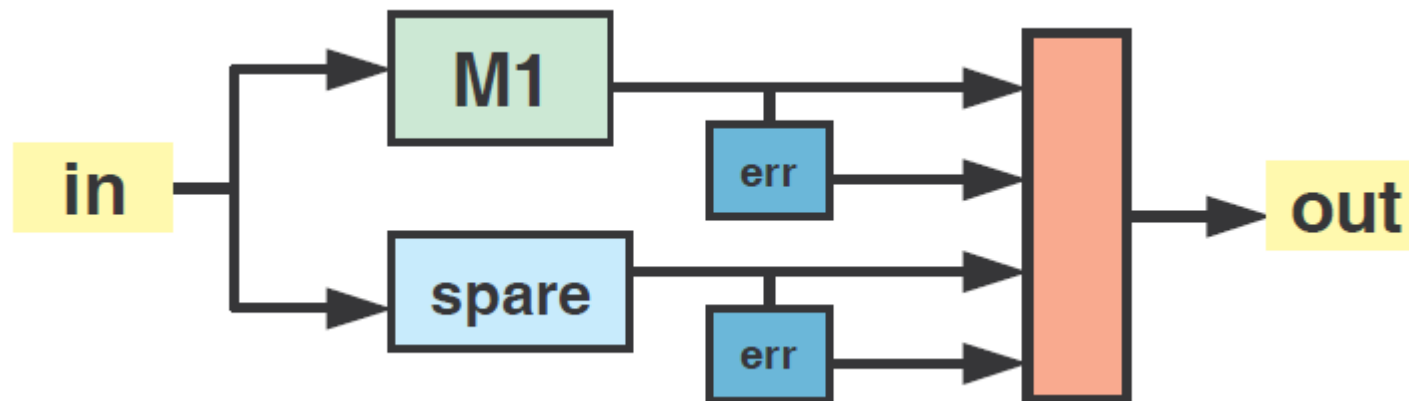
$$R_{min} = (1 + \lambda t)e^{-\lambda t}, t = ??$$

- Enačbo lahko rešimo numerično ali z razvojem v Taylorjevo vrsto

$$LT \approx \frac{\sqrt{2}}{\lambda} \sqrt{\ln \frac{1}{R_{min}}}, \text{ če je } \lambda t < 1$$

Vroča rezerva

- Rezervni sistem, ki je ves čas delovanja primarnega sistema aktiven.
- Uporablja se tam ,kjer je prekinitev zaradi zagona rezerve nezaželena – rezervni napajalni sistemi.

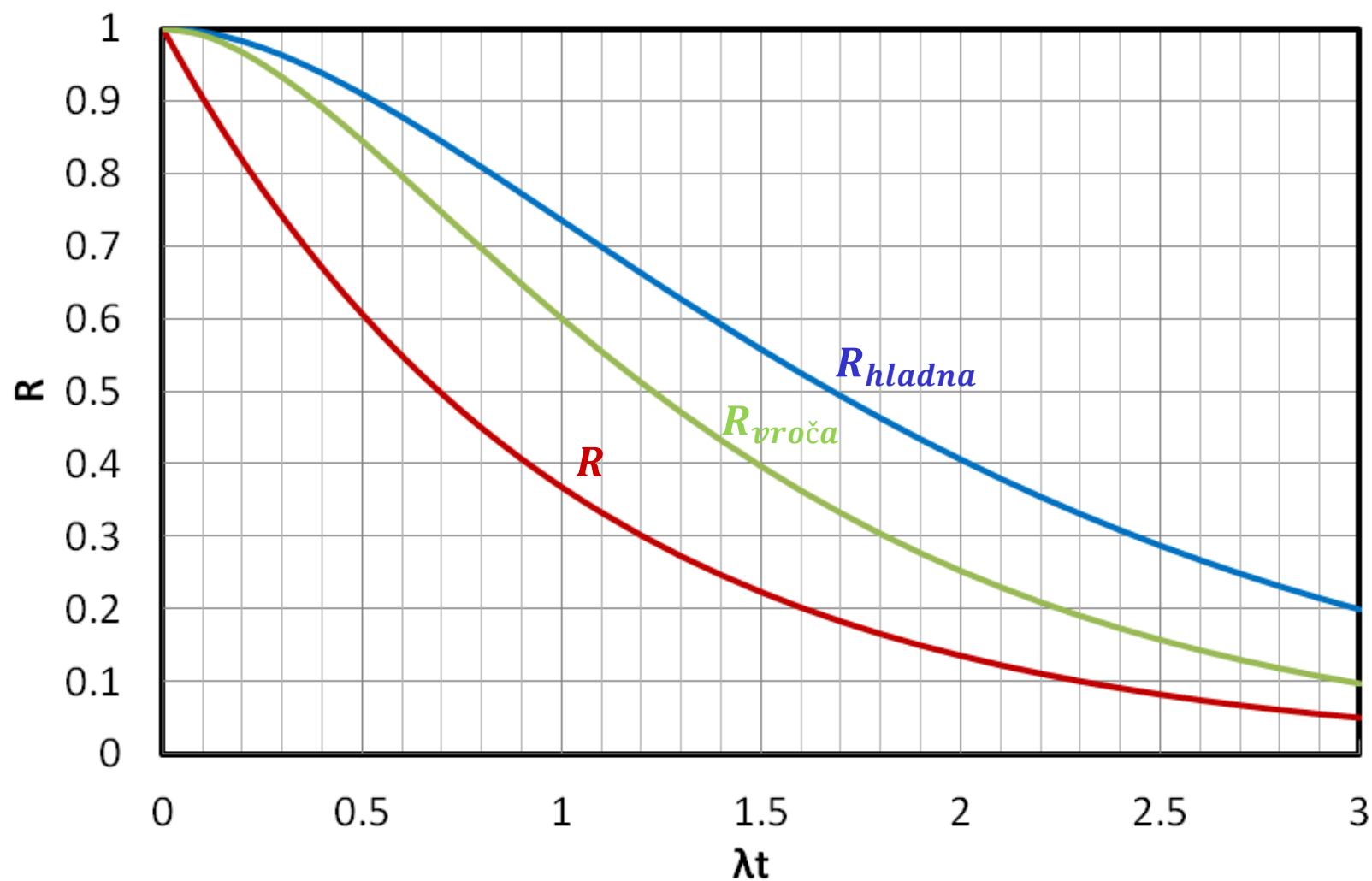


Zanesljivost vroče rezerve

- Oba sistema delujeta sočasno, torej zanesljivost s časom upada tudi rezervi.
- Zanesljivost sistema z vročo rezervo je enaka sistemu, ki ga sestavljata dva vzporedno povezana sklopa:

$$R_{sist}(t) = R_1(t) + R_2(t) - R_1(t)R_2(t)$$

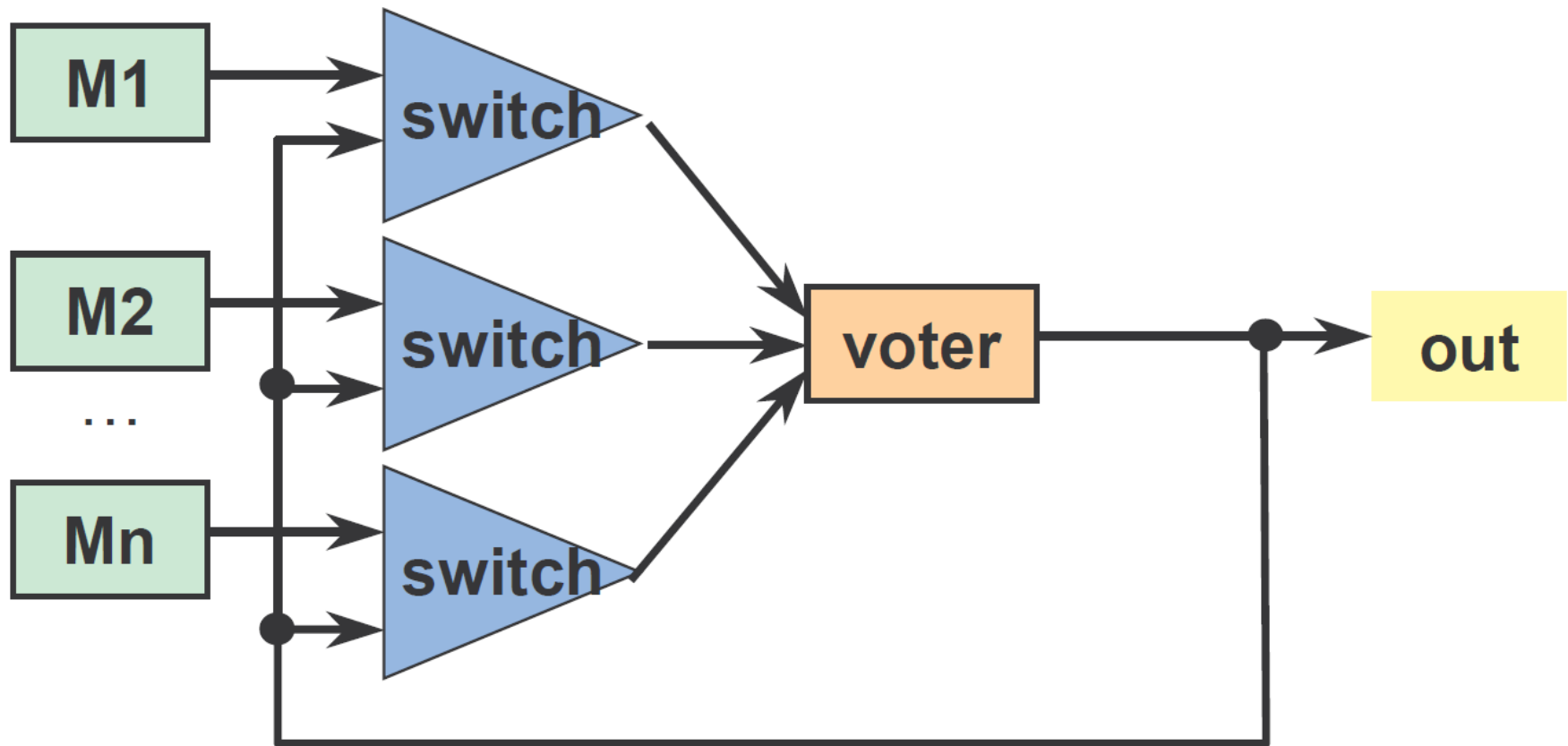
Primerjava rezerv



Hibridne redundance

- Kombinacija aktivne in pasivne redundance
 - Pasivna redundanca
 - Maskiranje napake
 - Aktivna redundanca
 - Detekcija, lokalizacija, omejitev in odprava
- Pristopi
 - Samo-izločujoča redundanca
 - N-modularna redundanca z rezervami
 - Trojno-dvojna redundanca

Samo-izločujoča redundanca



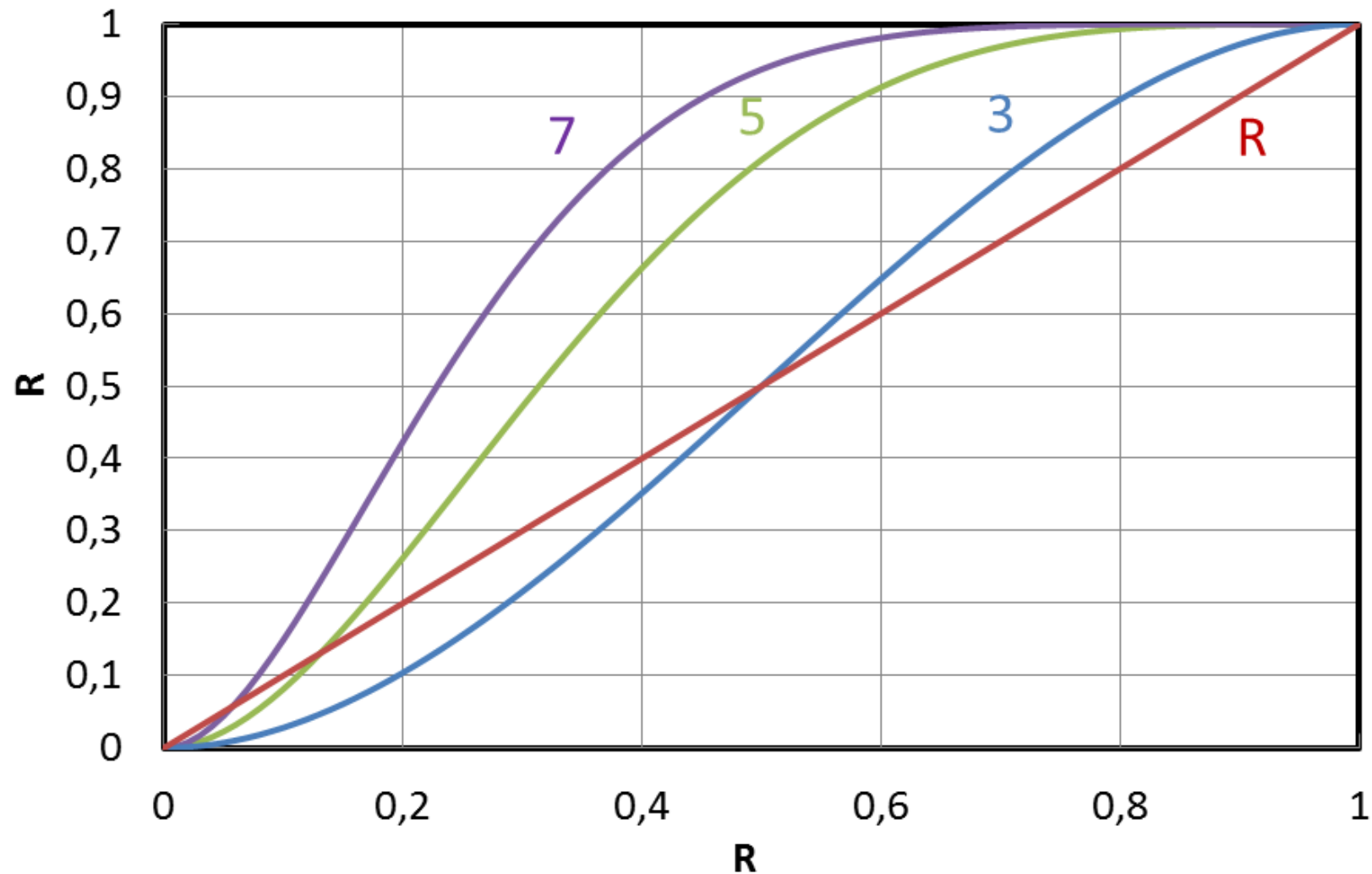
Lahko maskira $n-2$ napak

Zanesljivost samo-izločujoče redundance

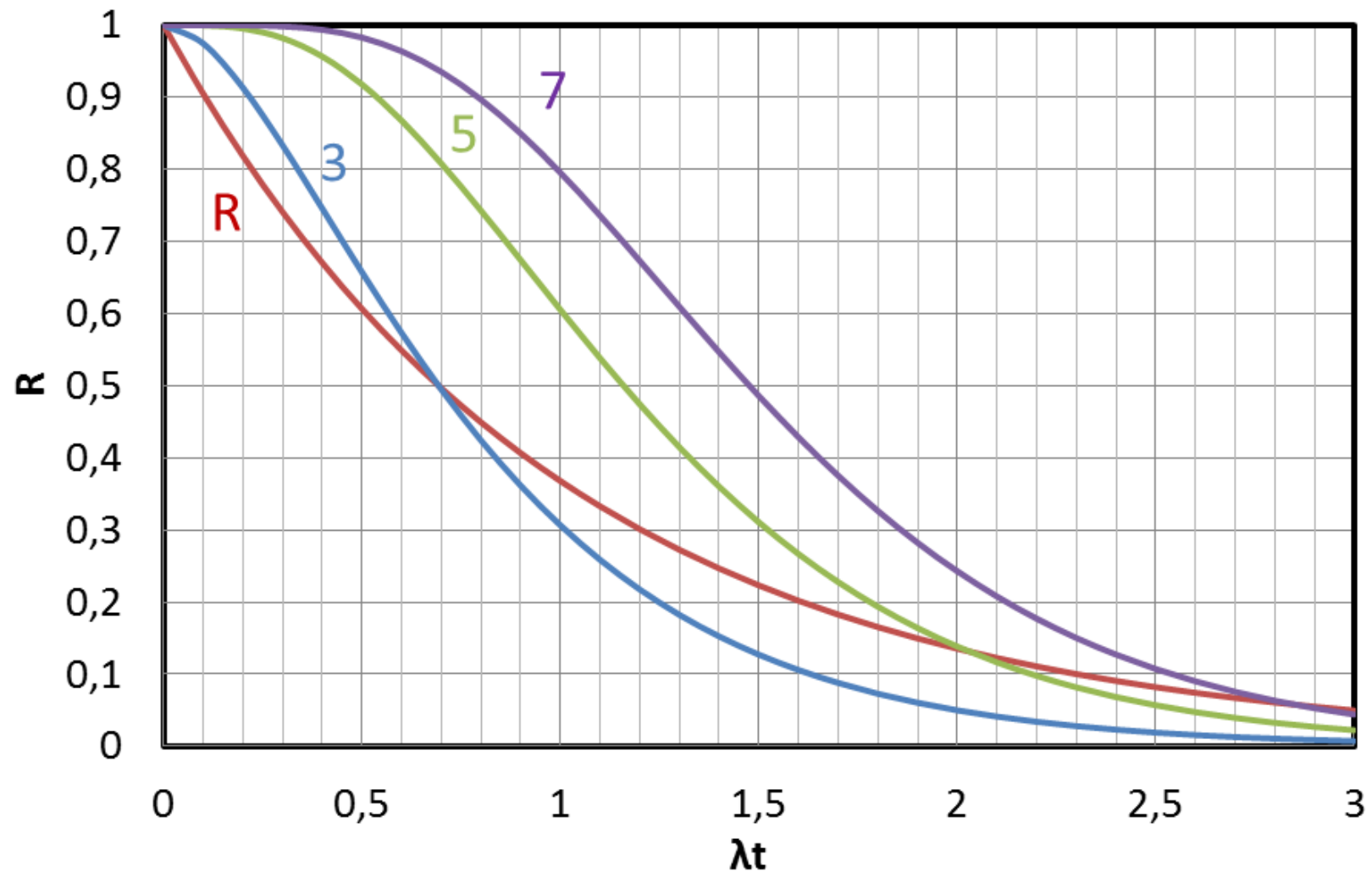
- Vsi moduli so delujejo vzporedno i njihove odpovedi so vzajemno neodvisne. Sistem deluje, če delujeta vsaj dva modula.
- Če so stikala in odločitveni blok idealna in je zanesljivost posameznih modulov enaka, potem sistem ne deluje,
 - če so vsi moduli odpovedali $(1 - R)^n$
 - če je ostal samo en delujoč modul $R(1 - R)^{n-1}$
- Ker je kombinacij, kjer en modul še deluje enako n dobimo

$$R_{syst} = 1 - [(1 - R)^n + nR(1 - R)^{n-1}]$$

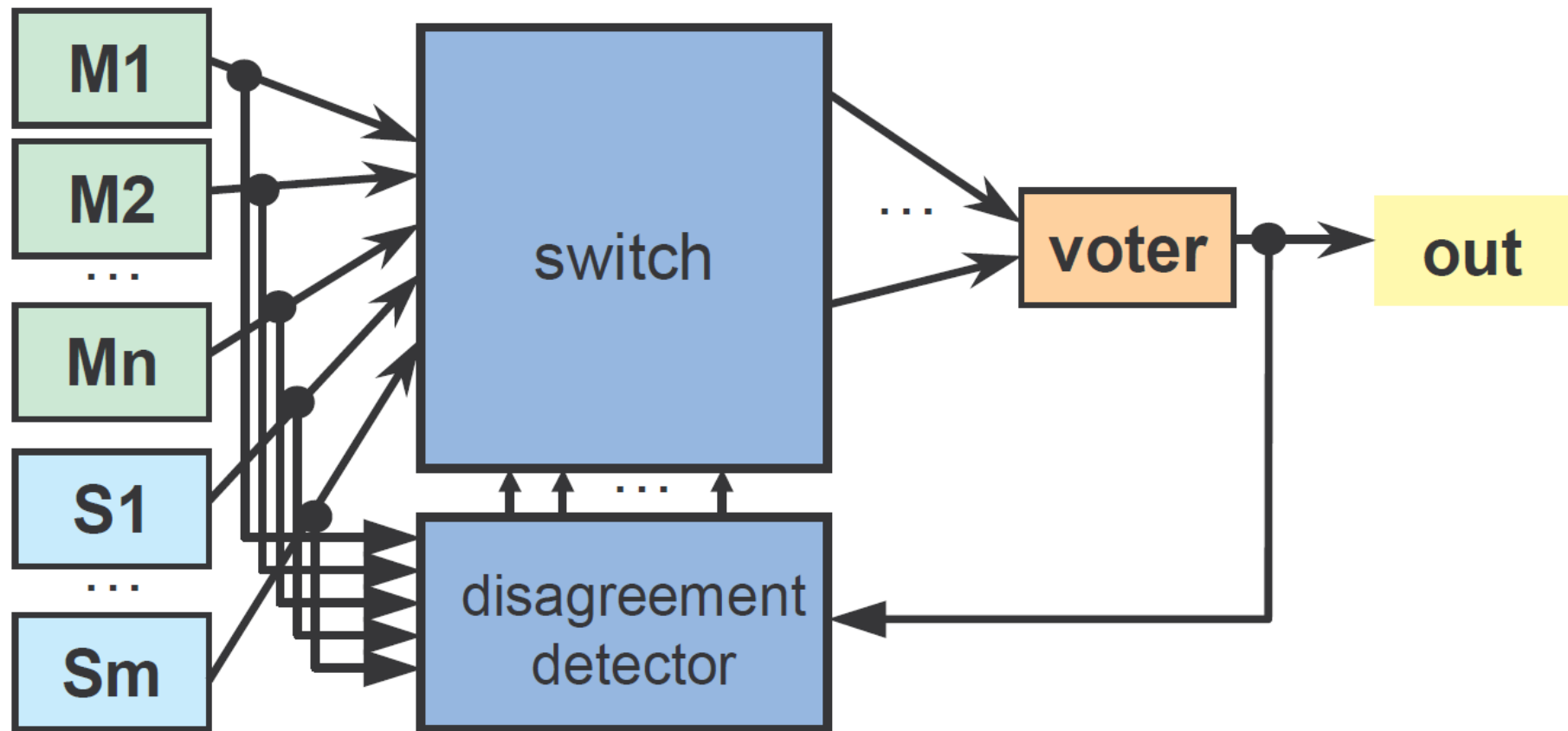
Zanesljivost samo-izločujoče redundance



Zanesljivost samo-izločujoče redundance



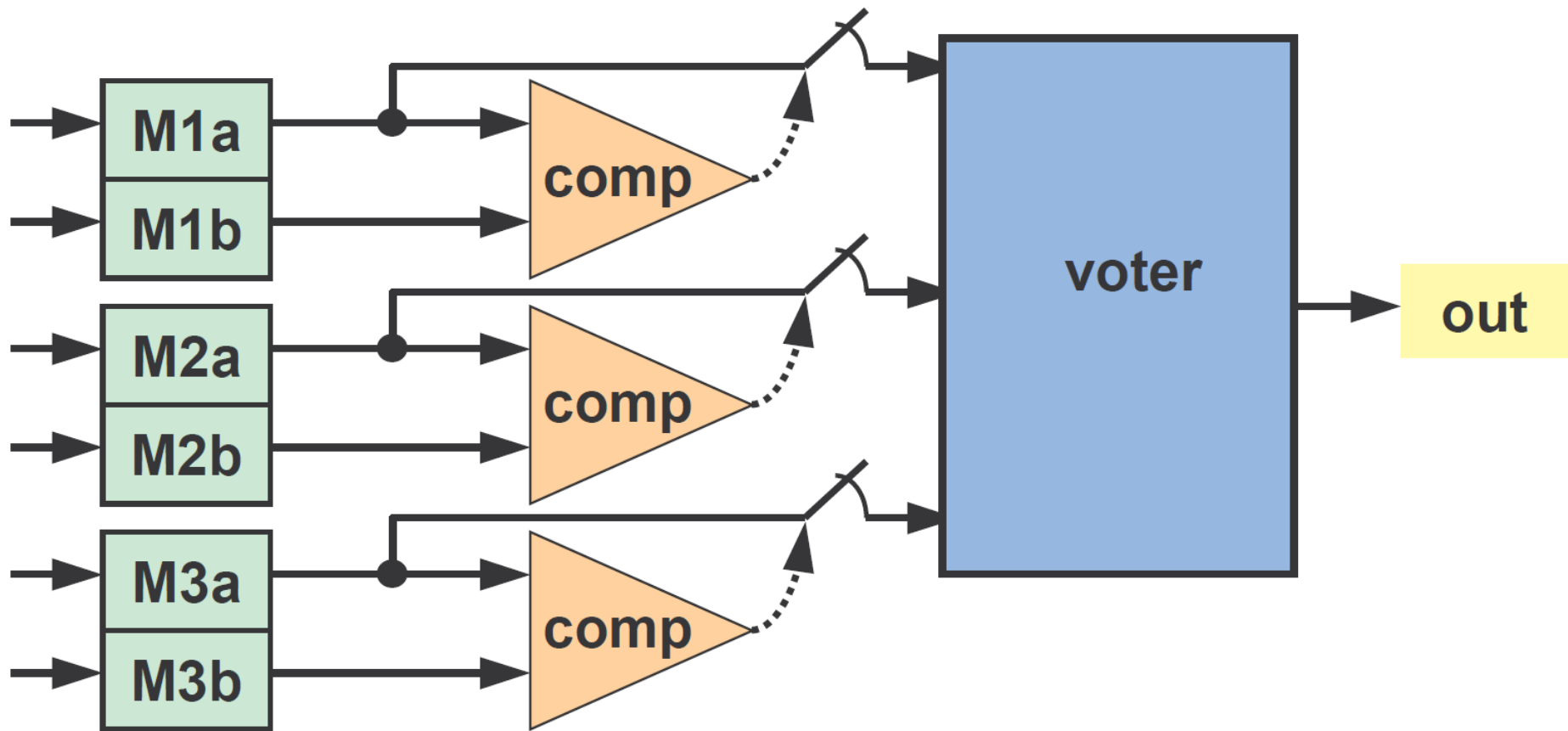
N-modularna redundanca z rezervami



N-modularna redundanca z rezervami

- Če katerikoli od glavnih modulov odpove, ga odklopi in zamenja z rezervnim
- Če so vsi rezervni moduli porabljeni, sta dve možnosti
 - Lahko izklopi „disagreement detektor“, in deluje naprej kot pasivna N -modularna redundanca
 - Lahko maskira največ $n/2 + m$ napak
 - Disagreement detektor ostane toda odločitveni sklop izloča okvarjene module iz odločitve.
 - Deluje kot samo-izločujoča redundanca z $n + k$ moduli in lahko maskira $n + k - 2$ napaki

Trojno-dvojna redundanca



Lahko maskira 2 napaki

Izbira sistema strojne redundance

- Računsko nezahtevni sistemi, visoka toleranca proti tranzientnim napakam
 - Pasivni ali hibridni redundančni sistemi
- Dolga življenjska doba, visoka razpoložljivost sistema
 - Aktivni redundančni sistemi
- Visoko kritične aplikacije, najvišja zahtevana zanesljivost
 - Hibridni redundančni sistemi

Informacijska redundanca



EMŠO

	D	D	M	M	L	L	L	R	R	Z	Z	Z	K
EMŠO	1	0	0	2	9	8	2	5	0	0	6	4	4
utež	7	6	5	4	3	2	7	6	5	4	3	2	

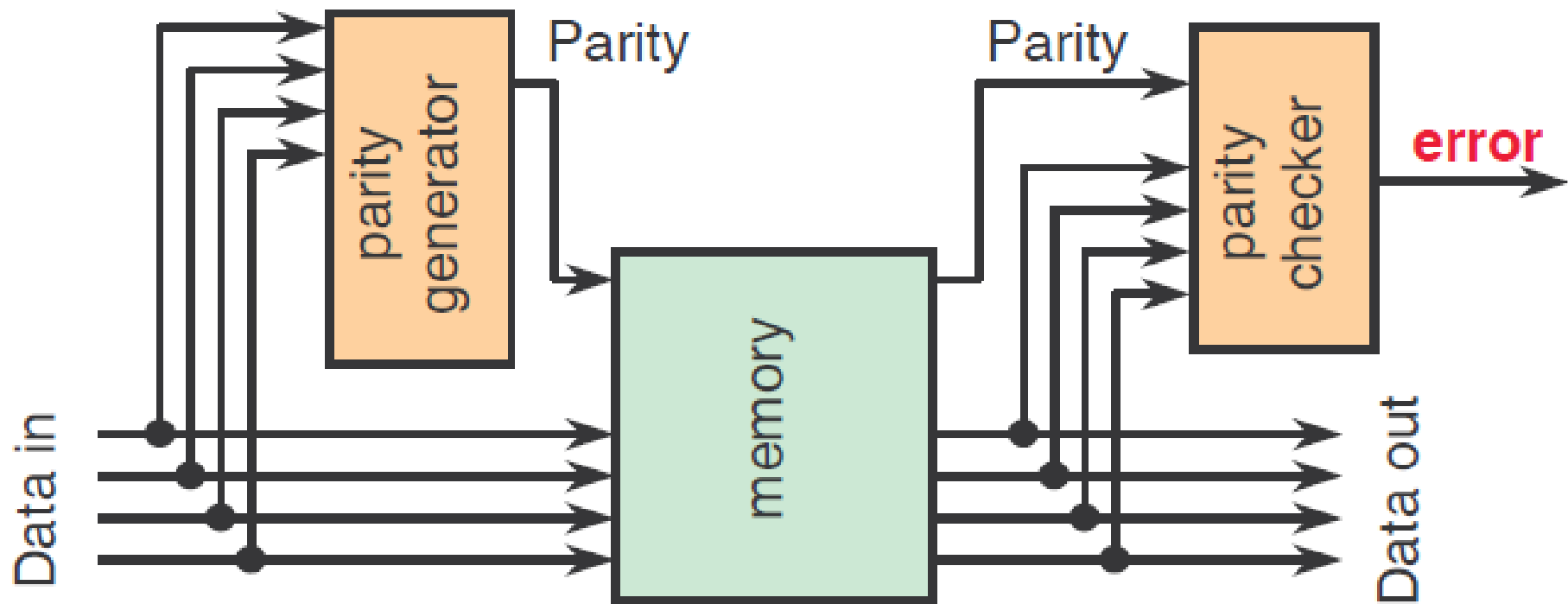
- DDMMLLL – datum rojstva
- RR – območje rojstva osebe
 - 00-09 – tujci
 - 10-19 – Bosna in Hercegovina
 - 20-29 – Črna gora
 - 30-39 – Hrvaška (33 - Zagreb)
 - 40-49 – Makedonija
 - 50-59 – Slovenija (uporabljena je samo 50)
 - 60-69 – (ni v uporabi)
 - 70-79 – Ožja Srbija (71 - Beograd)
 - 80-89 – Pokrajina Vojvodina (80 - Novi Sad)
 - 90-99 – Pokrajina Kosovo
- ZZZ – zaporedna številka rojstva
 - 000-499 za moške,
 - 500-999 za ženske
- K – kontrolna vsota
 - številke množimo z utežmi in seštejemo
 - delimo z 11 in izračunamo ostanek
 - če je ostanek = 0, potem K=0
 - sicer K = 11-ostanek
 - če je K = 10, se vzame naslednjo ZZZ

Paritetna koda

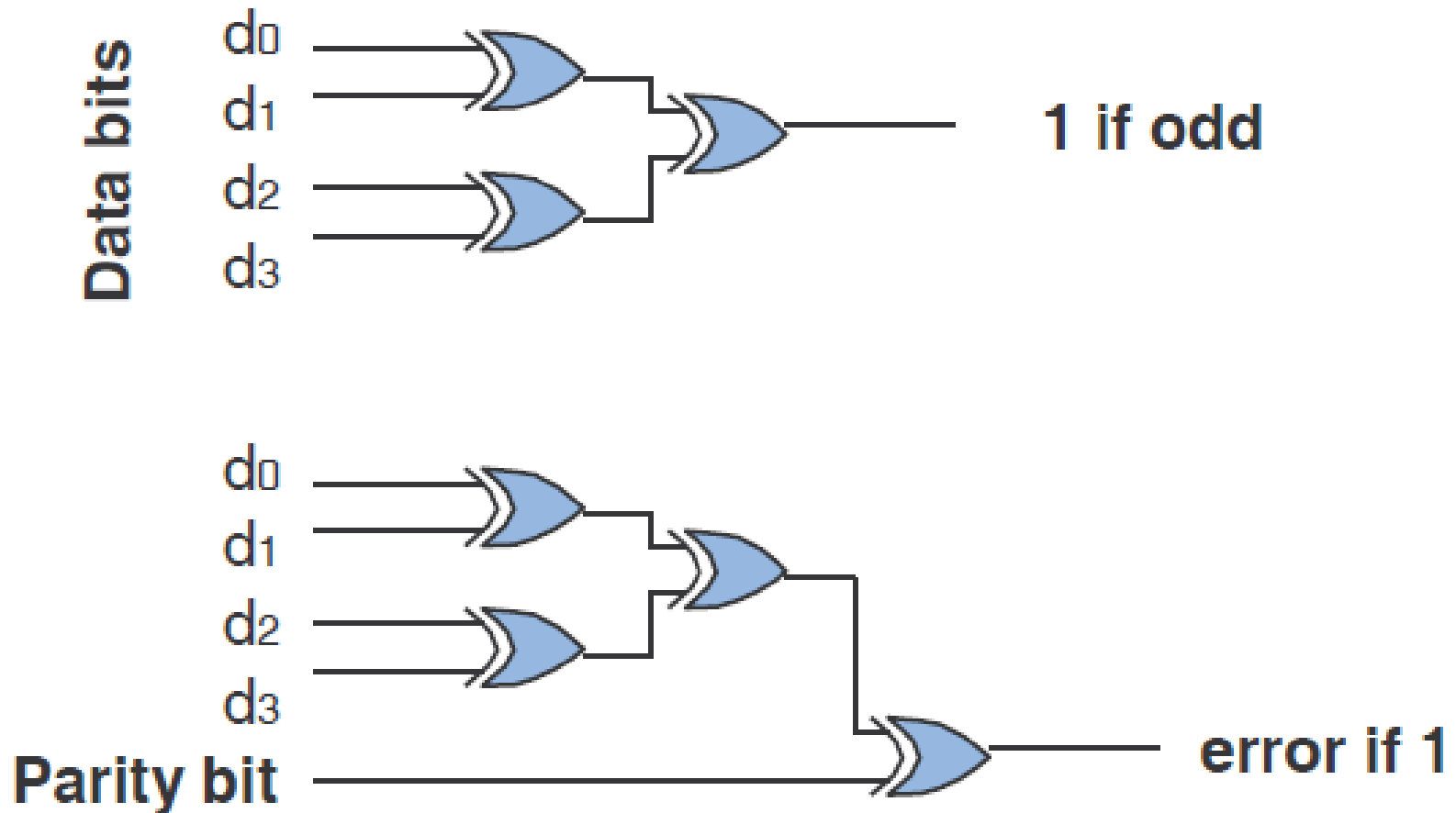
- Enobitna: Doda se 1 bit, ki pove, ali je število enic v podatku liho
 - soda pariteta: „1“, če je število enic v podatku liho
 - liha pariteta: „1“, če je število enic v podatku sodo

7 bitov podatkov	8 bit vključuje pariteto	
	Soda pariteta	Liha pariteta
000 0000	0000 0000	1000 0000
101 0001	1101 0001	0101 0001
110 1001	0110 1001	1110 1001
111 1111	1111 1111	0111 1111

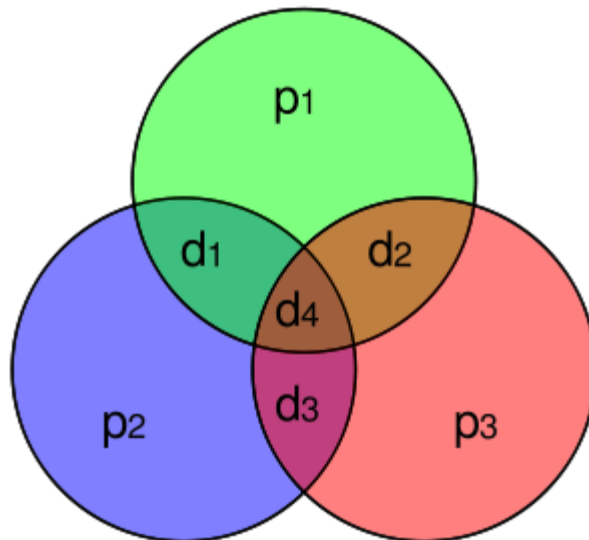
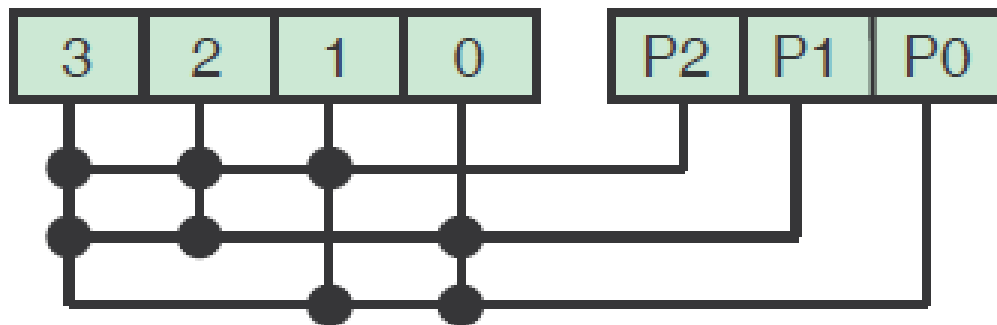
Primer uporabe paritete



Generator/preverjanje paritete



Prekrivajoče se paritetne kode (Hammingove kode)



Bit in error	Parity pattern
3	P2 P1 P0
2	P2 P1
1	P2 P0
0	P1 P0
P2	P2
P1	P1
P0	P0
no error	

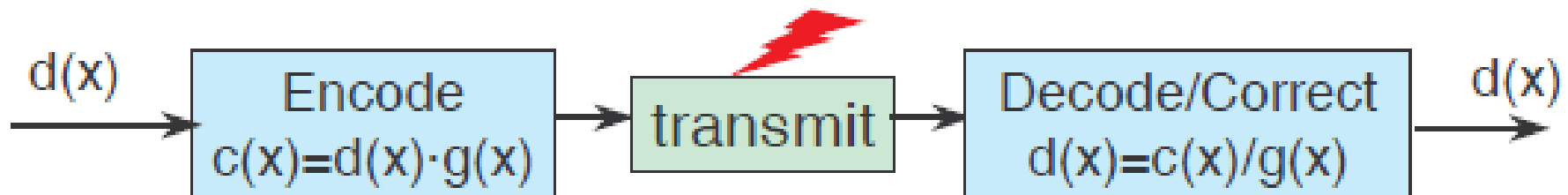
Redundanca hammingovih kod

- **k** podatkovnih bitov in **c** paritetnih bitov
- Če želimo detektirati vsako napako
 - $2^c \geq k + c + 1$

k	c	redundanca
2	3	150%
4	3	75%
8	4	50%
16	5	31%
32	6	19%
64	7	11%

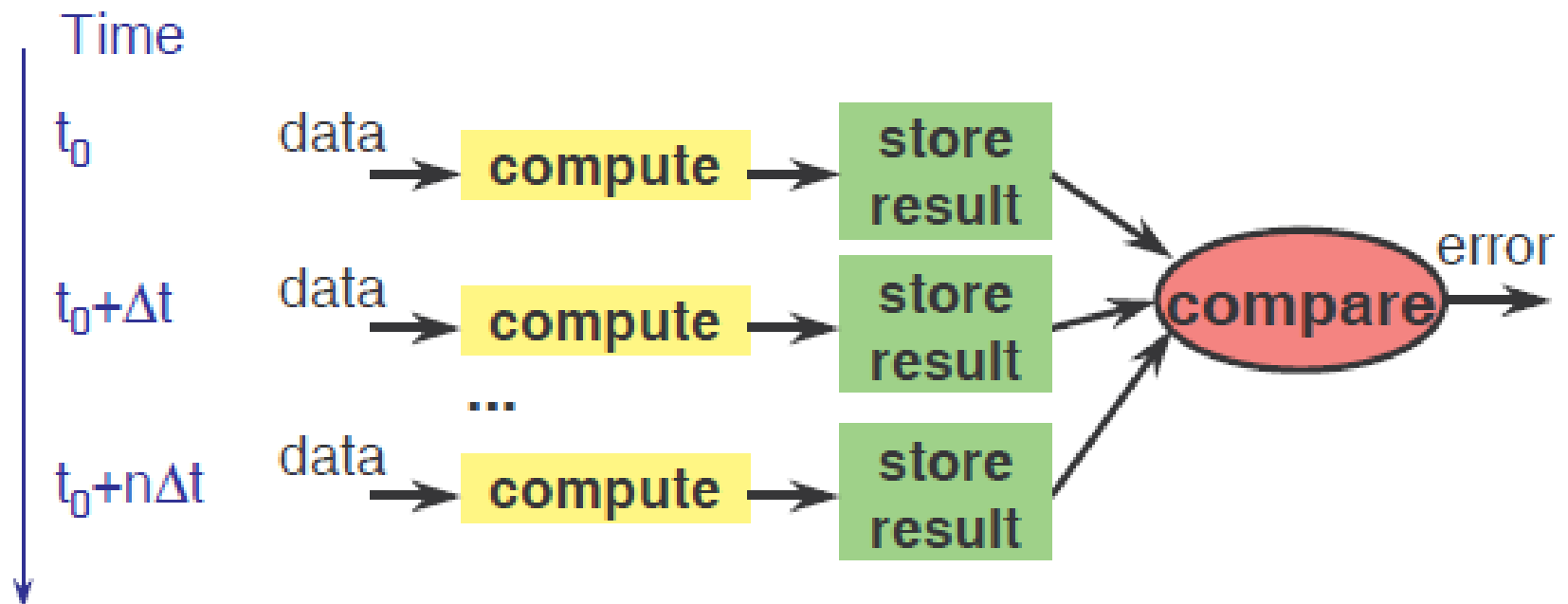
Ciklične kode

- Uporabljajo se tam, kjer se napake pojavljajo v skupinah
 - digitalne komunikacije, HDD, CD
- Dva tipa kod:
 - CRC (modemi, omrežni protokoli)
 - Reed-Solomon koda (CD, DVD, xDSL, DVBx, ...)

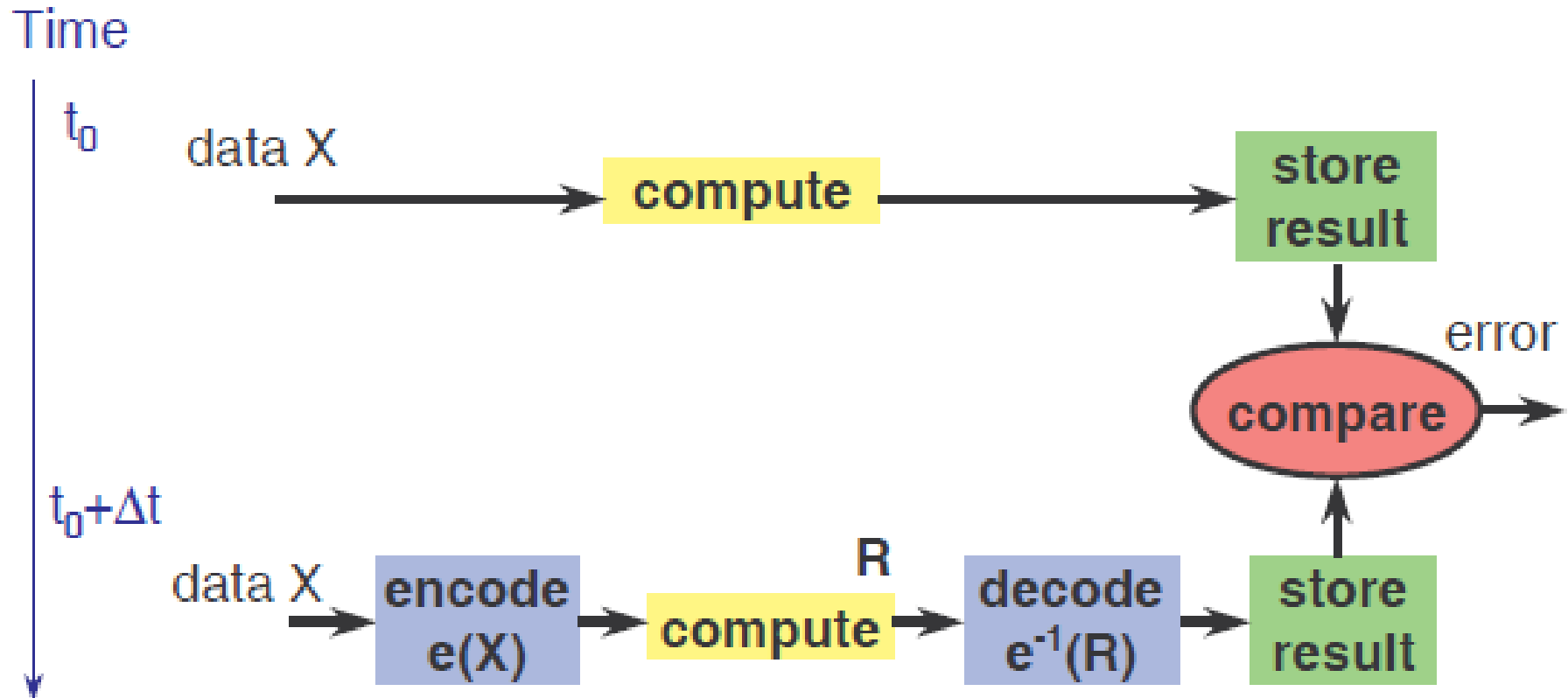


Časovna redundanca

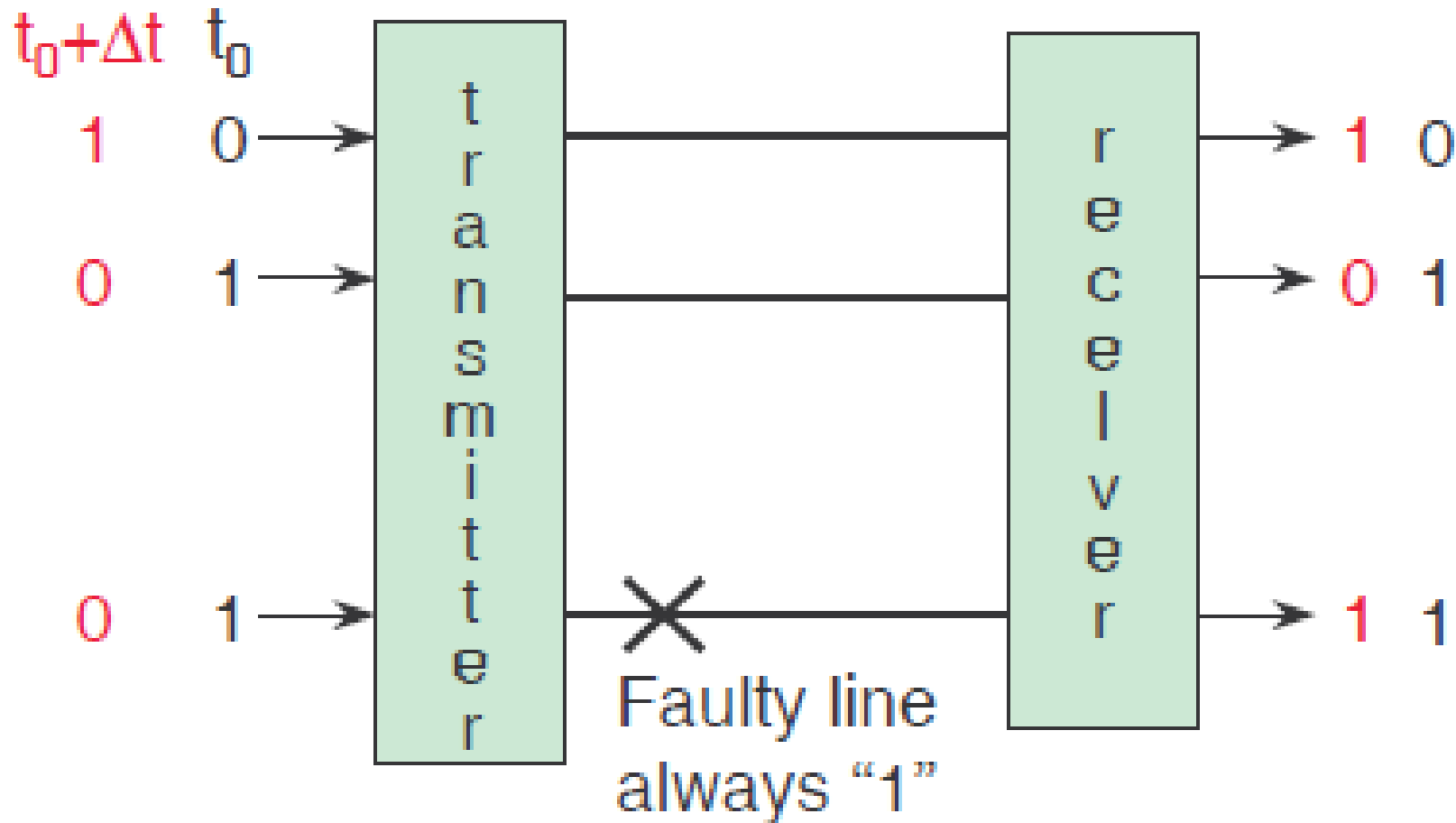
- Zmanjšanje potrebne dodatne strojne opreme na račun dodatnega potrebnega časa
- Detekcija tranzientnih napak



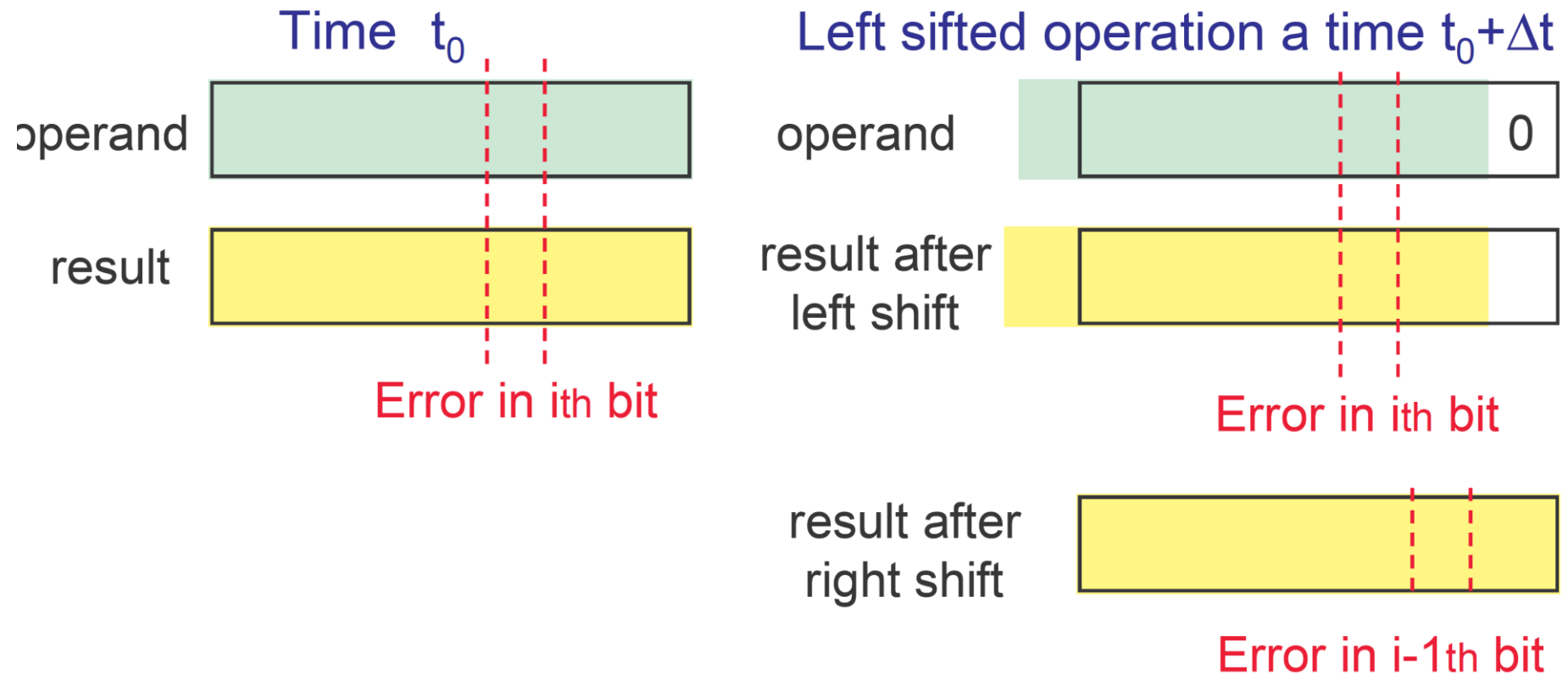
Detekcija permanentnih napak



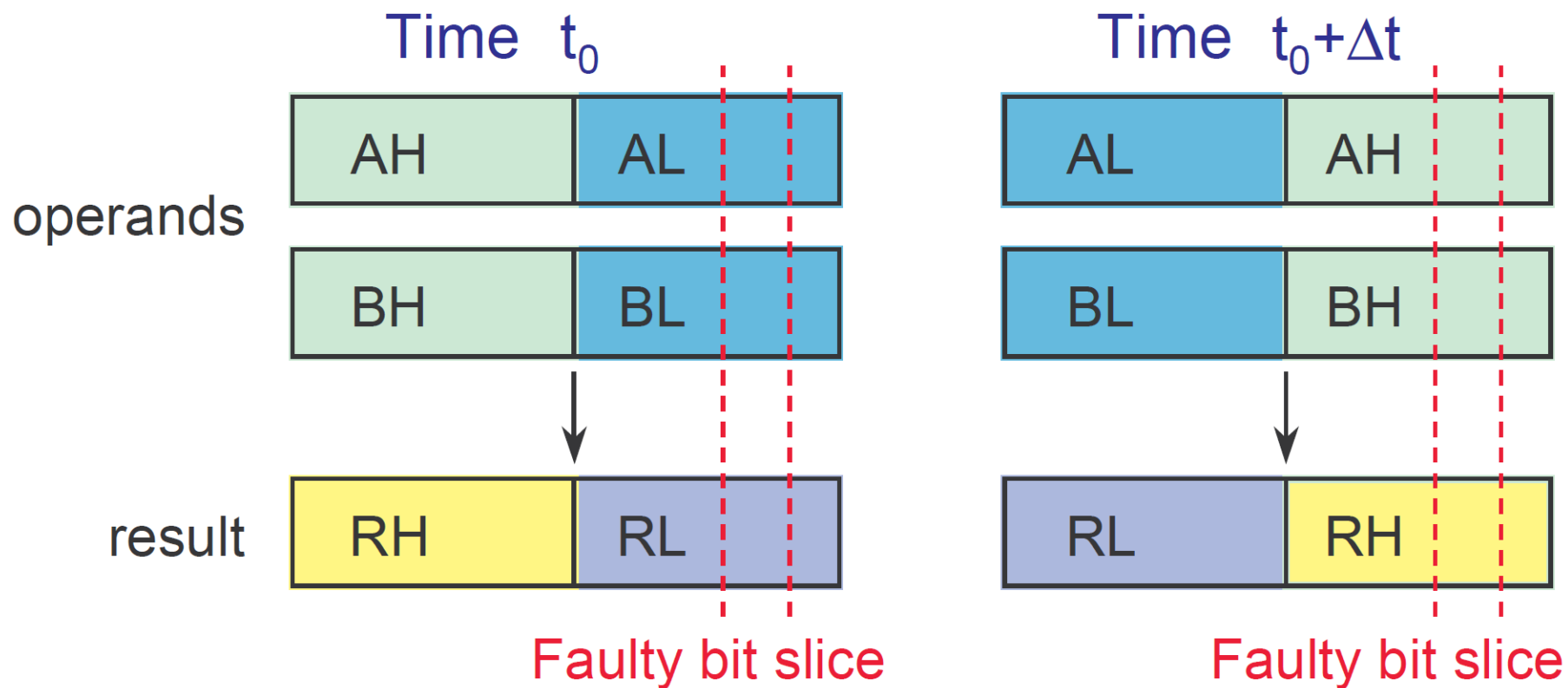
Časovna redundanca z alternirajočo logiko



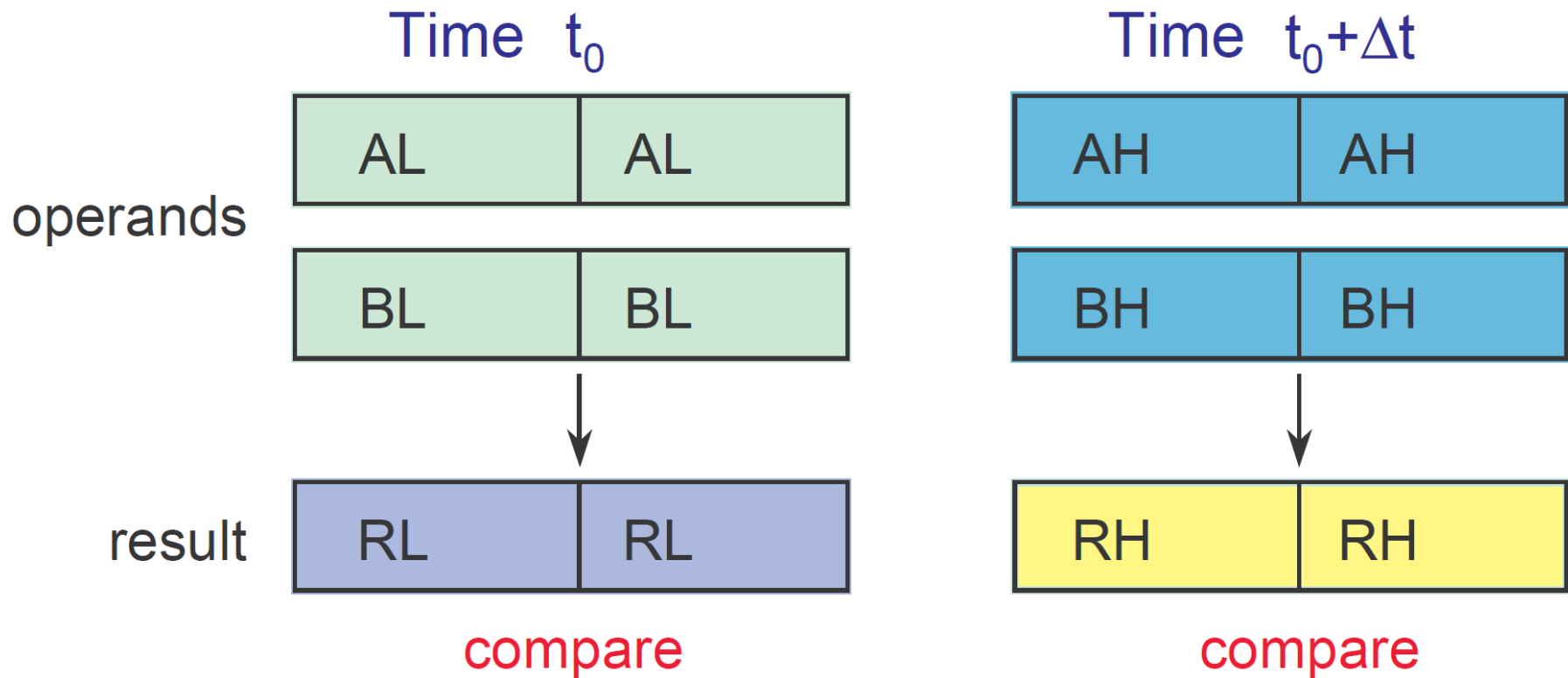
Časovna redundanca s pomikanjem operanda



Časovna redunanca s izmenjavo obeh polovic operanda



Časovna redundanca s podvojitvijo in primerjavo obeh polovic operandov



Programska redundanca

- Napake v programih se pojavijo
 - Dizajnerske napake
 - Slabo poznavanje programskega jezika
 - Matematično nekonsistentne funkcije
 - Slabe specifikacije programa
 - ne zajemajo vseh možnosti
 - Zaradi nepričakovanih vhodnih podatkov
 - Ne preverjamo in izločamo nelogičnih kombinacij vhodnih podatkov, ki bi povzročali napake
- Z enostavno pomnožitvijo in primerjavo modulov ne dosežemo učinka, saj vsi identični programski moduli dajejo identično (napačne) rezultate

Časovno preverjanje

- Nadzorni časovnik – „Watch dog timer“
- Predpostavka:
 - Sistem je brez napake če se periodično javlja
- WDT je časovnik, ki mora biti brisan v določenih intervalih s strani sistema
 - Sicer povzroči ponoven zagon sistema



Kodno preverjanje

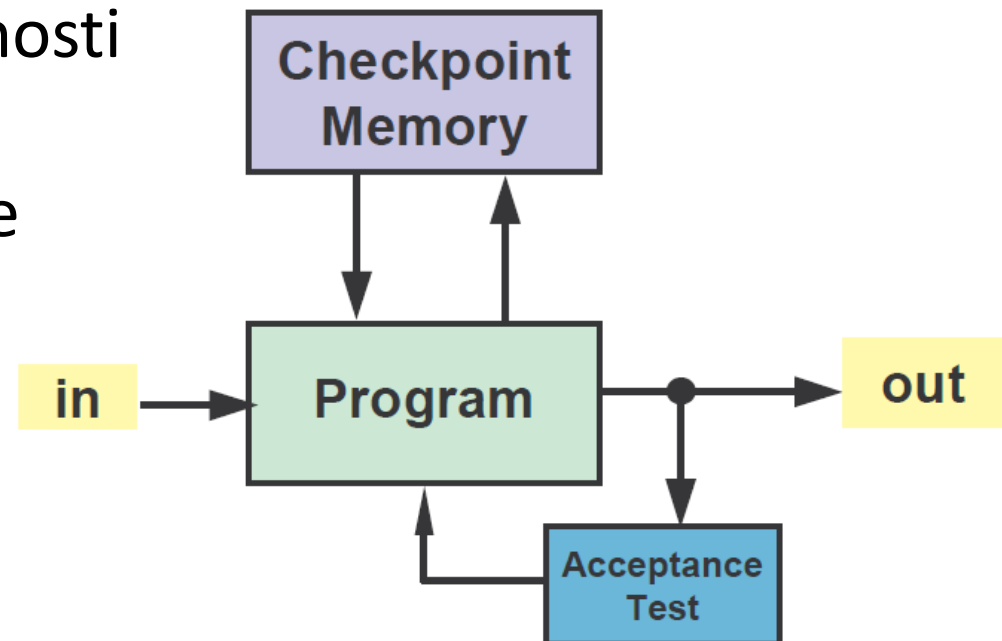
- Uporaba informacijske redundance
 - Prenos informacij med programskimi moduli
- Uporaba inverznih operacij za preverjanje izhodnih podatkov tako, da se jih prevede nazaj na vhodne
- Preverjanje smiselnosti rezultatov
 - Območje vrednosti
 - Ničelne vrednosti
 - Negativne vrednosti
- Strukturna preverjanja
 - Preverjanje znanih lastnosti struktur
 - Število elementov polja, dolžine tekstovnih nizov, naslovov kazalcev
 - Dodajanje redundance v podatkovne strukture
 - dolžina niza, dodatni kazalec na niz, ...

Izolacije napak

- Strukture in funkcije v sistemu naj imajo kup omejitev, ki definirajo katere akcije in podatki so dovoljeni v sistemu v nekem trenutku
 - Sistem se razdeli v module, ki so med sabo čim bolj neodvisni
 - Omejevanje komunikacije med moduli
 - Izogibanje skupnim virom informacij
 - Npr. izogibanje globalnim spremenljivkam in strukturam
 - Horizontalno razdeljevanje
 - Glavne funkcije programa razdelimo v neodvisne veje
 - Uporaba večnitnega programiranja (threads)
 - Komunikacijo med nitmi opravljajo kontrolni moduli
 - Vertikalno razdeljevanje
 - Razdelitev kontrole in procesiranja v hierarhijo od zgoraj navzdol
 - Visoko nivojski moduli se osredotočijo na kontrolo
 - Nizko nivojski moduli se ukvarjajo s procesiranjem

Poprava napak

- Upravljanje z izjemami
 - Prekinitev normalnega toka programa s funkcijo, ki upravlja z nepričakovano napako
- Shranitev stanja programa v pomnilnik
- Koda za preverjanje pravilnosti rezultata ob napaki sproži vrnitev na originalno stanje programa



Redundančni pristopi

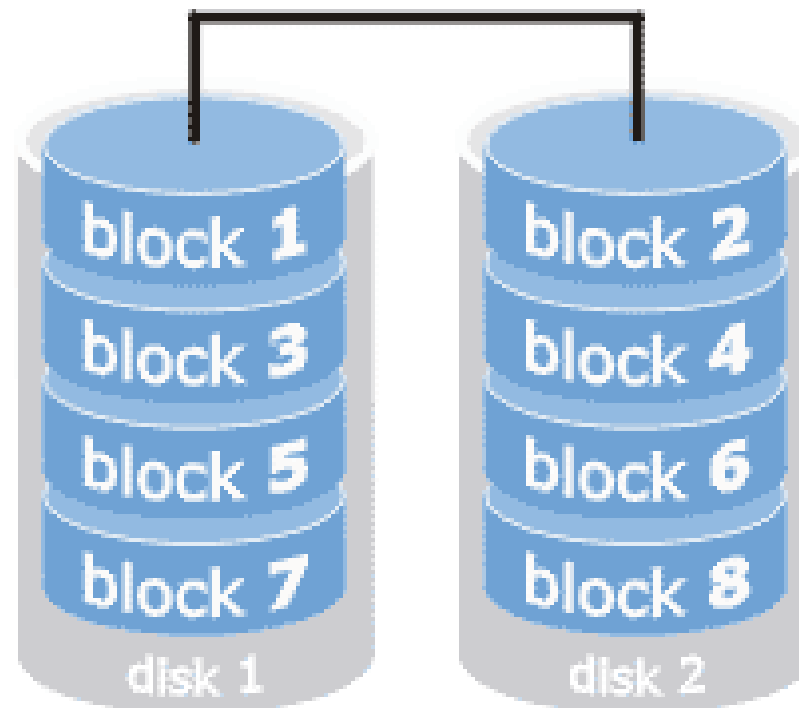
- Podvajanje procesorjev
 - Na več procesorjih teče ista koda.
- Diverzija kode
 - Več neodvisnih razvivalnih ekip
 - Različni programski jeziki
 - Različni algoritmi
- Rezultati se primerjajo
 - Pristopi podobni kot pri strojni redundanci (TMR, ...)

RAID – Redundant Array of Independent Disks



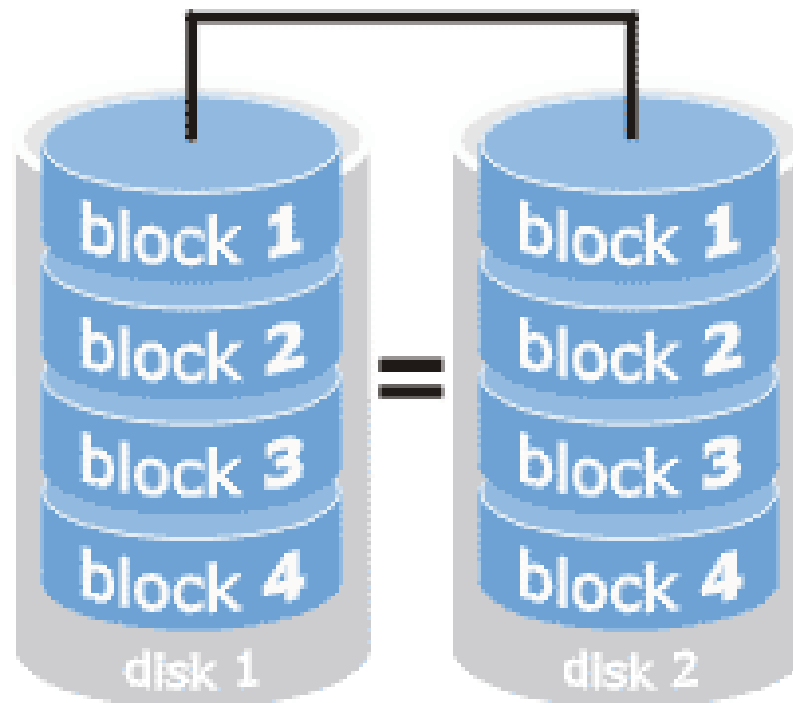
RAID 0

striping



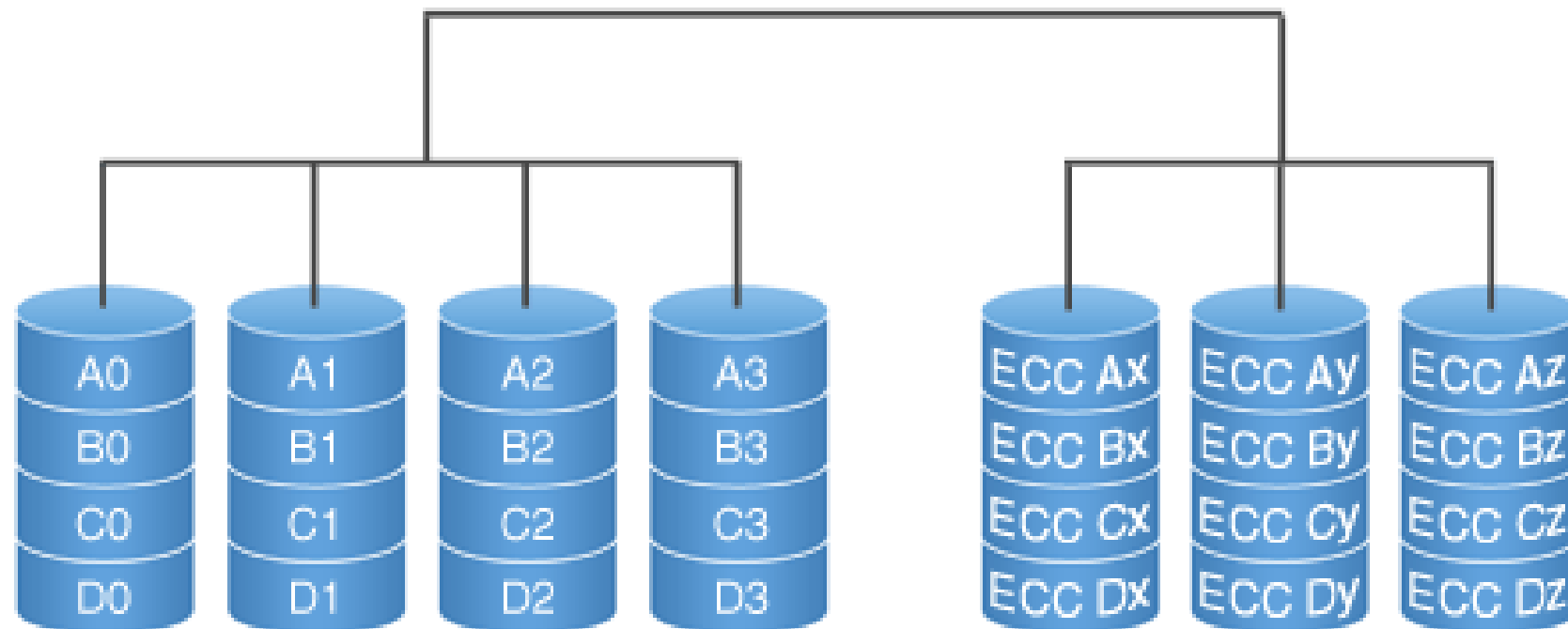
RAID 1

mirroring



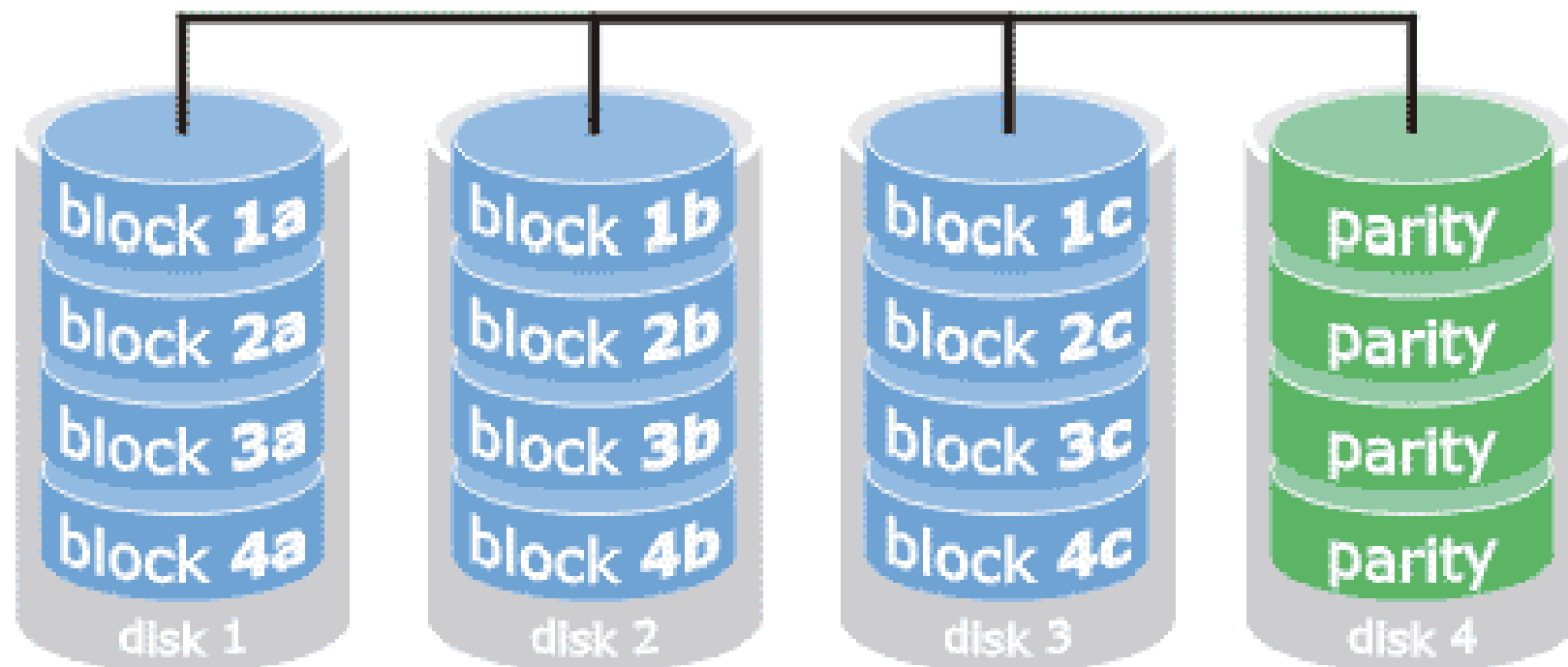
RAID 2

Hamming Code ECC



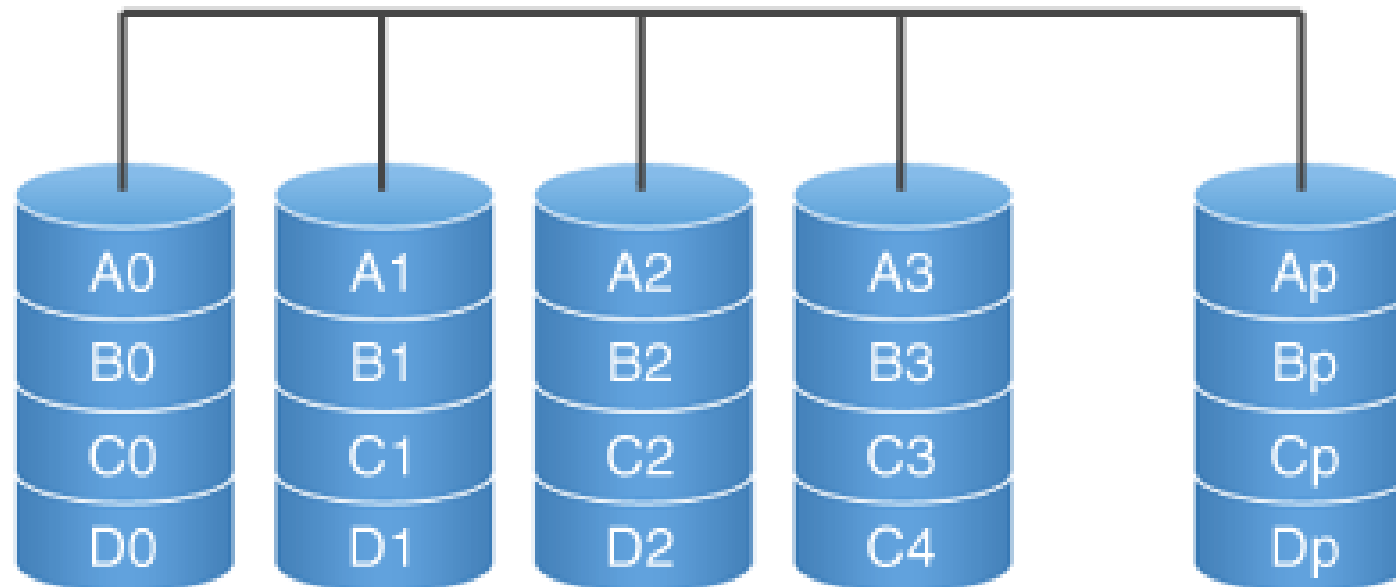
RAID 3

parity on separate disk



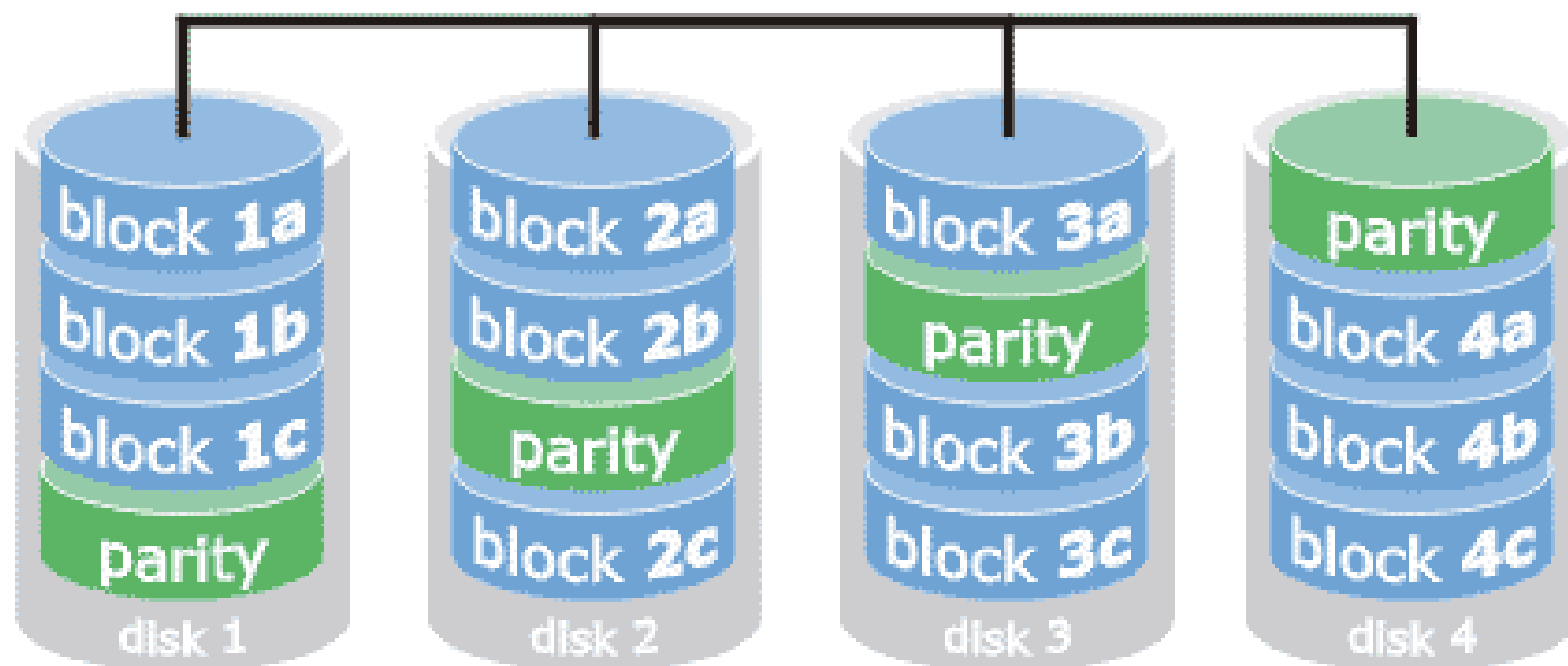
RAID 4

Independent Data Disks with Shared Parity Disk



RAID 5

parity across disks



RAID 0+1 (10)

