

# Zanesljivostna analiza ARM9 razvojnega sistema FRI – SMS

---

## Seminar pri predmetu RZD

*Jure Balabanič, Mitja Cerkvėnik, Jani Černe, Vlado Dimitrieski, Jernej Gorički, Jure Janša, Gregor Jeraj, Tine Kavčič, Urša Levičnik, Dimitar Mitrevski, Zahir Mujanović, Matej Peršolja, Darko Pevec, Miha Pinterič, Matija Polajnar, Matej Simčič, Sašo Skube, Matic Tovšak, Andrej Veber, Kaja Vidmar, Gašper Žejn*

V okviru predmeta Računalniška zanesljivost in diagnostika smo izdelali zanesljivostno analizo ARM9 razvojnega sistema FRI – SMS po različnih metodah. To poročilo vključuje poglavja o zanesljivosti strojne opreme, zanesljivosti programske opreme, metodo FTA(Fault Tree Analysis), FMEA (Failure Mode and Effects Analysis) ter Markovsko analizo.

## Kazalo

|  |    |
|--|----|
| Seminar pri predmetu RZD .....   | 1  |
| 1. Zanesljivost strojne opreme.....  | 3  |
| 1.1. Uvod.....   | 3  |
| 1.2. MIL .....   | 4  |
| 1.3. Telcordia standard .....  | 11 |
| 1.4. Komentar k zanesljivosti strojne opreme .....                               | 17 |
| 2. Zanesljivost programske opreme.....   | 18 |
| 2.1. Pomen analize zanesljivosti programske opreme.....                          | 18 |
| 2.2. Programska oprema ARM modula.....   | 18 |
| 2.3. Vir podatkov.....   | 18 |
| 2.4. Uporabljeni modeli .....  | 19 |
| 2.5. Linearni model .....  | 19 |
| 2.6. Rezultati .....   | 20 |
| 3. Metoda FTA.....   | 26 |
| 3.1. Vzroki za odpoved.....  | 26 |
| 3.2. Interpretacija rezultatov .....   | 37 |
| 4. FMEA.....   | 38 |
| 4.1. Opis .....  | 38 |
| 4.2. Zgodovina.....  | 38 |
| 4.3. Razvoj.....   | 38 |
| 4.4. Koristi in uporaba.....   | 39 |
| 4.5. Analiza sistema ARM.....  | 40 |
| 4.6. Zaključek .....   | 42 |
| 5. Markovska analiza.....  | 43 |
| 5.1. Markovska analiza splošno (kaj je, kratka predstavitev) .....               | 43 |
| 5.2. Markovska analiza in zanesljivost (prednosti, slabosti...) .....            | 43 |
| 5.3. Markovska analiza sistema z mikrokrmilnikom AT91SAM9620 v RELEX Studio..... | 44 |
| 5.4. Diskusija .....   | 48 |
| 6. Literatura.....   | 49 |

# 1. Zanesljivost strojne opreme

## 1.1. Uvod

Prvotni standard za zanesljivost je bil MIL-HDBK-217. Oblikovan je bil za zagotavljanje matematičnih modelov zanesljivosti za vse vrste elektronskih naprav. Standard je priznan po celem svetu, uporabljajo pa ga tako gospodarske družbe, kot tudi obrambna industrija. Včasih se označuje tudi kot MIL 217, MIL Handbook 217, MIL-217, MIL-217F, MIL-STD-217, ali MIL -HDBK-217E. Najnovejša revizija MIL-HDBK-217 je Revision F Notice 2, ki je izšla februarja 1995.

Standard oziroma model za zanesljivost, imenovan Telcordia, je bil prvotno razvit s strani AT & T Bell Labs. Glavna koncepta MIL-HDBK-217 in Telcordie sta sicer zelo podobna, vendar je Telcordia dodala možnost testiranja na terenu in v laboratorijih, s čimer je pridobila na priljubljenosti pri različnih komercialnih organizacijah. Najnovejša različica standarda je Telcordia Issue 2. Telcordia ponuja deset različnih kalkulacijskih metod, pri čemer je vsaka metoda narejena tako, da vzame v obzir več informacij.

Kot že rečeno, obstaja med standardoma Telcordia ter MIL-HDBK-217 precej razlik. Ena od pomembnejših razlik je t.i. množilnik. Telcordia standard je bil razvit za izračunavanje stopnje neuspeha v številu FIT (failures in time – napak v času). Ta vrednost je izražena kot število napak na milijardo obratovalnih ur. Pri standardu MIL-HDBK-217 je ta vrednost izražena kot število napak na milijon obratovalnih ur. Druga pomembna razlika je v določanju ravni kakovosti posameznih delov. Telcordia podpira štiri standarde ravni kakovosti, pri čemer so ravni kakovosti enake za vse tipe komponent. MIL-HDBK-217 standard ponuja drugačen pristop, saj se ravni kakovosti razlikujejo od komponente do komponente. Iz tega vidika je Telcordia precej enostavnejša. Kar se tiče izračunov, sta standarda Telcordia ter MIL-HDBK-217 v osnovi zelo podobna. Je pa res, da pri primerjavi izračunov med modeloma ugotovimo, da so izračuni po standardu Telcordia bolj optimistični kot pa tisti pri MIL-u. Poleg tega izračuni pri Telcordii zahtevajo manj parametrov delov pri komponentah.

Znano je, da je model Telcordia hiter in enostaven za uporabo. Kot takšen je idealen za uporabo v različnih komercialnih aplikacijah, kar je bil tudi primaren cilj razvoja. To je tudi razlog, da se večina družb, ki proizvajajo računalnike, telekomunikacijske sisteme, medicinske sisteme ter napajalnike, odločijo za Telcordia standard. Po različnih raziskavah sodeč, naj bi Telcordia dala veliko bolj realistične rezultate, kot MIL, vendar pa se je potrebno zavedati, da oba standarda pokrivata svoje specifično področje zahtev. Omeniti velja še, da ima Telcordia pet okoljskih klasifikacij (3 za kopno, 1 za zrak in 1 za vesolje), medtem ko ima MIL 14 klasifikacij (3 za kopno, 8 za zrak, 1 za vesolje in 2 za vodo). Iz tega sledi, da če imamo produkt, ki se uporablja v vodi ali v zraku, je standard MIL boljše izbira.

## 1.2. MIL

Priročnik podpira uporabo dveh različnih metod določevanja intenzivnosti odpovedovanja sistemov: *part stress analysis prediction* in *parts count prediction*. Prva metoda se uporablja v končni fazi, ko je sistem že postavljen in natančno specificiran, druga pa se uporablja v začetni fazi razvoja sistema, ko še nimamo natančnih podatkov o posameznih komponentah, potrebujemo pa približne ocene.

Ker je naš sistem že postavljen smo za naše potrebe uporabili prvo metodo, torej *part stress analysis prediction*. Za potrebe te metode smo morali najprej določiti komponente, poiskati njihove specifikacije ter na podlagi tega poiskati intenzivnosti odpovedovanja posameznih komponent, konektorjev in tiskanega vezja.

Pri tej nalogi pa smo naleteli na nekaj problemov. Večina komponent na naši razvojni ploščici je znanih (razvidnih iz oznake na čipu) in se dokaj hitro ugotovi kakšnemu namenu služijo. Ko pa želimo natančneje preveriti specifikacije, se izkaže, da zelo hitro ostanemo brez podatkov, ki jih potrebujemo za izračun intenzivnosti odpovedovanja po MIL standardu. Najtežje je priti do podatkov o kvalitetnih razredih komponent. Prav tako večina proizvajalcev nič ne pove o samem ohišju, tako da je treba na tip ohišja sklepati glede na videz. Kljub skupnim informacijam pa se še vedno da priti do končnih izračunov. Nekaj komponent je vseeno ostalo neznanih in smo jim dodali verjetnost odpovedi  $1 \text{ Failure}/10^6 \text{ Hours}$ .

Večji problem predstavlja letnica priročnika. Od leta 1991, ko je bil izdan sta pretekli že skoraj dve desetletji in v tem času so se nekatere značilnosti komponent drastično spremenile. Komponente so postale manjše, spremenili so se načini izdelave, pojavile so se nove komponente, ki jih MIL-HDBK-217F sploh ne predvideva in tudi velikosti spomina so se znatno povečale (tukaj je bilo odstopanje največje, zato smo vzeli kar največjo velikost, ki nam jo je ponudil MIL). Tako je bilo potrebno ob računanju intenzivnosti odpovedovanja malce iznajdljivosti. Naj še opozorimo, da se lahko intenzivnosti odpovedovanja zelo razlikujejo, če bi primerjali s sodobnimi izračuni. Zato raje poleg samih števil (ki lahko zavajajo) navedemo še enačbe po katerih smo računali intenzivnosti odpovedovanja.

Intenzivnost odpovedi celotnega sistema izračunamo tako, da seštejemo intenzivnosti odpovedovanja posameznih komponent. V našem primeru smo dobili intenzivnost odpovedi celotnega vezja:

$$\lambda_p = 783.8679004 \text{ Odpovedi}/10^6 \text{ ur}$$

V nadaljevanju so zapisane enačbe komponent po katerih smo računali, podana je še tabela intenzivnosti odpovedovanja posameznih komponent, ter sliki sprednje in zadnje strani razvojnega sistema.

### 1.2.1. Enačbe elementov

Ne glede na to za katere komponente računamo intenzivnost odpovedovanja ( $\lambda_p$ ) se ponavljajo nekateri parametri:

- $\pi_E$  – Environment Factor: v kakšnem okolju se uporablja naprava. Za vse komponente na razvojni ploščici smo predvideli da je okolje zemeljsko mobilno, saj se ploščica zaradi namena uporabe veliko prenaša
- $\pi_Q$  – Quality Factor: kakšne kvalitete je komponenta. Ker tega ne vemo, vedno uporabimo najslabši faktor.
- $\pi_L$  – Learning Factor: koliko časa je komponenta v izdelavi. Ker je ta razvojna ploščica že nekaj časa na trgu, predvidevamo, da so tudi elementi že več kot 2 leti v proizvodnji.

#### Mikrovezja (microcircuits, microprocessors):

$$\lambda_p = (C_1\pi_T + C_2\pi_E) \pi_Q\pi_L \text{ odpovedi}/10^6 \text{ ur}$$

- $C_1$  – faktor kompleksnosti vezja (depends on no. of bits, if bipolar or MOS)
- $\pi_T$  – temperature factor – dobimo ga iz tabele, ali pa ga je potrebno izračunati iz enačbe, glede na to ali je mikrovezje silicijevo ali GaAs:

$$\pi_T = \exp\left(\frac{-E_a}{8.617 \cdot 10^{-5} \left(\frac{1}{T_J + 273} - \frac{1}{298}\right)}\right) \text{ za silicijevo in } \pi_T = \exp\left(\frac{-E_a}{8.617 \cdot 10^{-5} \left(\frac{1}{T_J + 273} - \frac{1}{423}\right)}\right) \text{ za GaAs}$$

V tej enačbi se pojavita zopet dva nova parametra in sicer  $E_a$  ter  $T_J$ .  $E_a$  je Activation energy,  $T_J$  pa Junction Temperature oz. temperatura spoja.

$T_J$  se izračuna po enačbi  $T_J = T_C + P\theta_{JC}$ , pri čemer je  $T_C$  Case Temperature,  $P$  Device Power Dissipation in  $\theta_{JC}$  Junction to case thermal resistance.  $E_a$  pa dobimo iz tabele, s pomočjo  $T_J$ .

Vidimo, da je pri tem parametru največ računanja, še posebno, če ne poznamo temperature spoja.

- $C_2$  – Package Failure Rate – kako vpliva ohišje na odpoved vezja

#### Spomin (microcircuits, memory)

$$\lambda_p = (C_1\pi_T + C_2\pi_E + \lambda_{CYC}) \pi_Q\pi_L \text{ odpovedi}/10^6 \text{ ur}$$

Vidimo, da je enačba zelo podobna enačbi za mikroprocesorje. Nenazadnje spadajo tako pomnilniki, kot mikroprocesorji v isto poglavje MIL-a. Enačbi se razlikujeta le po parametru  $\lambda_{CYC}$ . Ta parameter upoštevamo le pri ocenjevanju Flotax in Textured-Poly EEPROM, sicer je 0, in ne vpliva na zanesljivost.

$$\text{Parameter izračunamo tako: } \lambda_{CYC} = \left[ A_1 B_1 + \frac{A_2 B_2}{\pi_Q} \right] \pi_{ECC}$$

( $A_1$  in  $A_2$  parametra odvisna od največjega števila programirnih ciklov v življenju EEPROMa  $B_1$  in  $B_2$  parametra odvisna od  $T_J$ ,  $\pi_{ECC}$  pa je odvisen od tega kakšen Error Correction Code je na čipu)

## Konektorji

Vse konektorje v našem vezju obravnavamo enako in po isti enačbi:

$$\lambda_p = \lambda_b \pi_K \pi_P \pi_E \text{ odpovedi}/10^6 \text{ ur}$$

- $\lambda_b$  – Base Failure Rate: odvisna od materiala konektorja in temperaturnih lastnosti le-tega. Ker konektorji niso podrobneje specificirani, da bi lahko vedeli iz kakšnega materiala so in kakšne so njihove temperaturne lastnosti, smo vzeli najslabši možni faktor.
- $\pi_K$  – kakšna je frekvenca pretikanja kablov
- $\pi_P$  – število aktivnih kontaktov

## Kondenzatorji

Kondenzatorji v vezju so dveh vrst. Za 3 kondenzatorje smo ugotovili, da so elektrolitski, zato smo jih uvrstili v skupino *Capacitors, Fixed, Electrolytic, Tantalum, Solid*, čemur ustreza enačba:

$$\lambda_p = \lambda_b \pi_{CV} \pi_{SR} \pi_Q \pi_E \text{ odpovedi}/10^6 \text{ ur}$$

- $\pi_{CV}$  – Capacitance Factor, odvisen od velikosti kondenzatorja
- $\pi_{SR}$  – Series Resistance Factor, odvisno od upornosti vezja (ohm/volt)

Druga vrsta kondenzatorjev je tipa SMD, nismo pa izvedeli podrobnosti teh elementov, zato smo vzeli kar enačbo skupine *Capacitors, Fixed, Ceramic, General Purpose*:

$$\lambda_p = \lambda_b \pi_{CV} \pi_Q \pi_E, \text{ pri čemer so parametri že znani}$$

## Upori

Uporov v vezju je prav tako več različnih vrst, vsi pa imajo enako enačbo:

$$\lambda_p = \lambda_b \pi_R \pi_Q \pi_E \text{ odpovedi}/10^6 \text{ ur}$$

- edini še neznan parameter v tej enačbi je  $\pi_R$  – Resistance Factor – odvisen od velikosti upora

## LED Dioda

Spada pod skupino *Photodetectors, Opto-isolators, Emitters* in ima enačbo:

$$\lambda_p = \lambda_b \pi_T \pi_Q \pi_E \text{ odpovedi}/10^6 \text{ ur}$$

Vsi faktorji v tej enačbi so znani že iz razlag prejšnjih enačb.

## Dioda

Za 4 diode smo ugotovili, da so *Schottky Diode*, ki imajo enačbo:

$$\lambda_p = \lambda_b \pi_T \pi_S \pi_C \pi_Q \pi_E \text{ odpovedi}/10^6 \text{ ur}$$

- $\pi_S$  – Electrical Stress Factor
- $\pi_C$  – Contact Construction Factor

## Tranzistor

Predpostavljamo, da spada v skupino *Transistors, Low Frequency, Bipolar*, saj ne vemo podrobnosti o njemu. Enačba:

$$\lambda_p = \lambda_b \pi_T \pi_A \pi_R \pi_S \pi_Q \pi_E \text{ odpovedi}/10^6 \text{ ur}$$

- $\pi_A$  – Application Factor
- $\pi_R$  – Power Rating Factor
- $\pi_S$  – Stress Factor

### Tipka

Spada v skupino *Switches – Toggle and Pushbutton* in ima enačbo:

$$\lambda_p = \lambda_b \pi_{CYC} \pi_L \pi_C \pi_E \text{ odpovedi}/10^6 \text{ ur}$$

Še neznani parametri tukaj so:

- $\lambda_b$  – Base Failure Rate tukaj odvisna od kvalitete in opisa elementa
- $\pi_{CYC}$  – Cycling Factor – kolikokrat na uro se gumb pritisne
- $\pi_L$  – Load Stress Factor – obremenitev elementa
- $\pi_C$  – Contact Form and Quantity Factor

### Tiskano vezje

Pri tiskanem vezju poznamo dve različici: pri eni upoštevamo, da je tiskano vezje z metaliziranimi izvrtinami, drugo pa *Printed Circuit Board*. Prvo ima enačbo:

$$\lambda_p = \lambda_b (N_1 \pi_C + N_2 (\pi_C + 13)) \pi_Q \pi_E \text{ odpovedi}/10^6 \text{ ur}$$

- $N_1$  – Quantity of Wave Soldered Functional PTHs
- $N_2$  – Quantity of Hand Soldered PTHs
- $\pi_C$  – Complexity Factor

Druga različica pa enačbo:

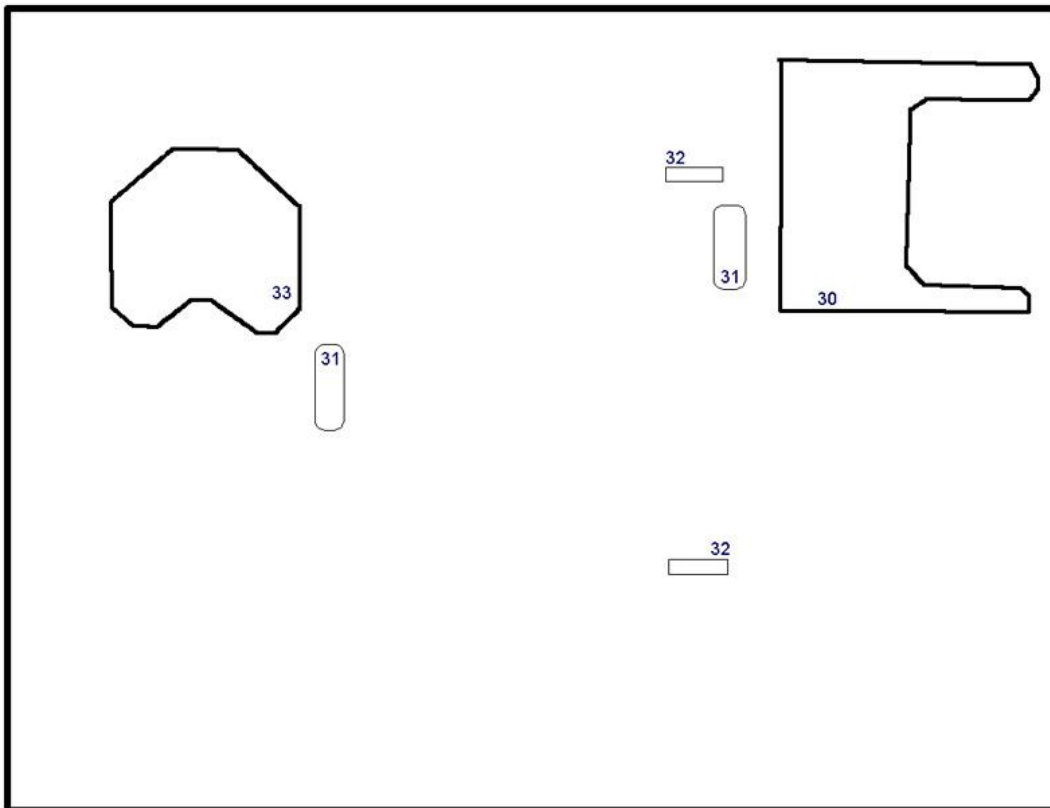
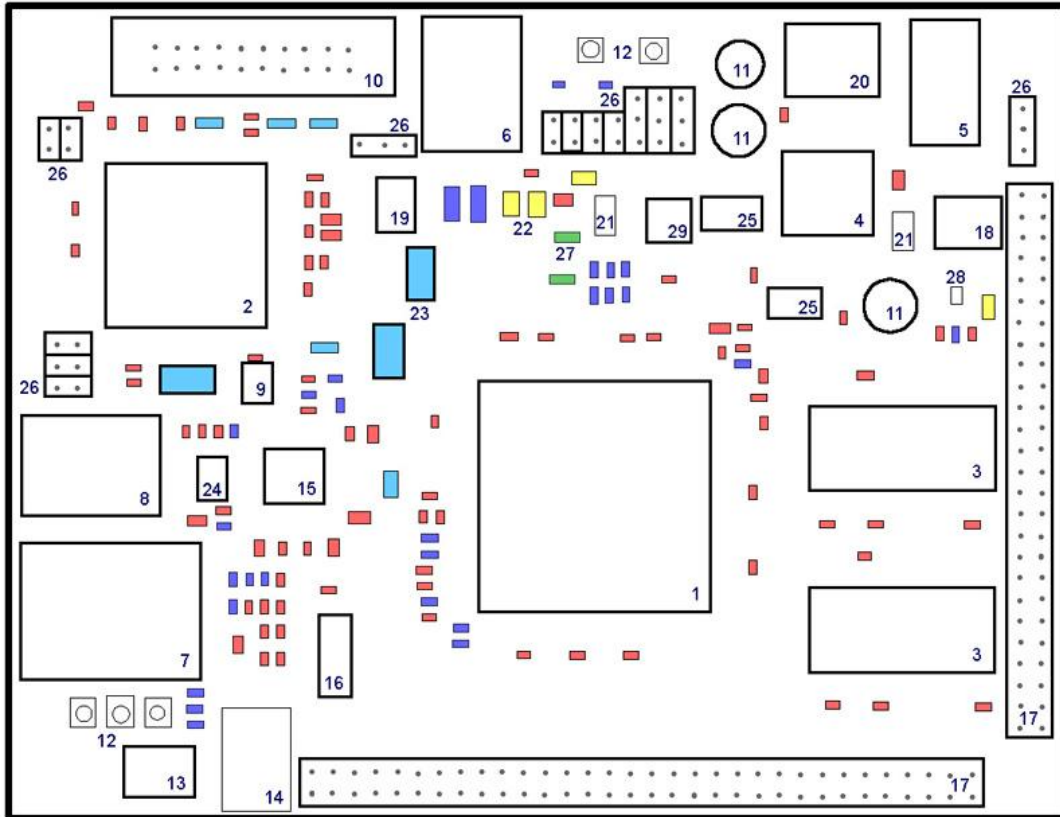
$$\lambda_p = \lambda_b \pi_K \pi_P \pi_E \text{ odpovedi}/10^6 \text{ ur}$$

- $\pi_K$  – Mating/Unmating Factor
- $\pi_P$  – Active Pins

Naše vezje ima nekaj metaliziranih izvrtin, drugače pa je PCM s SMD elementu, zato vzamemo drugo enačbo za izračun.

| Št. | Element   | odpovedi/10 <sup>6</sup> ur | opombe                  |
|-----|---|-----------------------------|-------------------------|
| 1   | ARM Procesor  | 0.916144                    |                         |
| 2   | Reprogrammable Connectivity Processor                       | 0.760004                    |                         |
| 3   | 256Mbit sinhroni RAM  | 5.213984                    | 2x                      |
| 4   | Low Dropout Positive Regulator                              | 0.05376                     |                         |
| 5   | Power konektor (pwrjack)                                    | 6.216                       |                         |
| 6   | USB konektor  | 6.216                       |                         |
| 7   | LAN konektor  | 6.216                       |                         |
| 8   | USB konektor  | 6.216                       |                         |
| 9   | 34164, PWIX3 (MC34614 – beli tisk)                          | 1                           | neznan                  |
| 10  | JTAG konektor   | 24.864                      |                         |
| 11  | Elektrolitski kondenzator                                   | 3.36864                     |                         |
| 12  | LED   | 0.129536                    | 5x                      |
| 13  | Reset tipka   | 123.264                     |                         |
| 14  | RS232 konektor  | 9.9456                      |                         |
| 15  | 10/100 Fast Ethernet Physical Layer Single Chip Transceiver | 0.23656                     |                         |
| 16  | MAX3232CTransceiver   | 0.1476                      |                         |
| 17  | Razširitveni pini   |                             |                         |
|     | – 64 pinov  | 80.37835008                 |                         |
|     | – 52 pinov  | 61.86374544                 |                         |
| 18  | DF 005S 718A  | 1                           | neznan                  |
| 19  | Low Voltage Bidirectional Transceiver                       | 0.1459864                   |                         |
| 20  | Power Connector   | 6.216                       |                         |
| 21  | Overcurrent protection device                               | 1                           | neznan                  |
| 22  | Schottky diode  | 6.048                       | 4x, Slika: rumeno       |
| 23  | Surface mount resistor                                      |                             | Slika: svetlo modro     |
|     | – 103   | 0.612                       | 6x                      |
|     | – 470   | 0.51                        | 2x                      |
| 24  | Chip ceramic resonator                                      | 1                           | ni v MIL                |
| 25  | Low dropout linear regulator                                | 0.00426                     | 2x                      |
| 26  | “jumperji”  |                             |                         |
|     | – 4 pini  | 10.5672                     |                         |
|     | – 6 pinov   | 12.432                      |                         |
|     | – 3 pini  | 9.9456                      |                         |
|     | – 17 pinov  | 22.3776                     |                         |
|     | – 3 pini  | 9.9456                      |                         |
| 27  | Transistor  | 0.87912                     | 2x, Slika: zeleno       |
| 28  | Micropower, 150mA Low-Dropout CMOS Voltage Regulator        | 0.07837872                  |                         |
| 29  | atmel 32 megabit dataflash                                  | 20.3536                     |                         |
| **  | Ostali manjši elementi na prvi strani:                      |                             |                         |
|     | – kondenzatorji   | 3.591                       | 84x, Slika: rdeče       |
|     | – upori   | 0.12672                     | 29x, Slika: temno modro |
| 30  | Čitalec SD kartic z 2GB SD kartico                          | 20.3536                     |                         |
| 31  | Chip ceramic resonator                                      | 1                           | 2x, ni v MIL            |
| 32  | srebrni element, 2 nožici                                   | 1                           | 2x, neznan              |
| 33  | baterija  | 1                           | ni v MIL                |
|     | Tiskano vezje - PCM   | 40.12                       |                         |





### 1.2.2. Komentar k rezultatom

Intenzivnost odpovedi glede na izračun bo precej velika – MTBF je približno 2 meseca. Največ k nezanesljivosti sistema doprinese tipka, ki ima intenzivnost odpovedi kar  $\lambda_p = 123.264$  odpovedi/ $10^6$  ur, ter konektorji. Odpoved tipke je kar razumljiva, saj je ta del mehanski in kar precej uporabljan. Nismo pa pričakovali take intenzivnosti odpovedovanja pri konektorjih. Res pa je, da smo pri konektorjih vedno vzeli najslabšo vrednost iz MIL-a, saj nismo dovolj natančno vedeli podatkov o teh elementih. Prav tako smo predpostavili, da so vsi pini, ki so na teh konektorjih aktivni, kar še dodatno doprinese k slabšemu rezultatu. Kot smo pričakovali, smo pri mikrovezjih (procesor in nekaj ostalih elementov) dobili precej majhno intenzivnost odpovedi, prav tako pri pasivnih elementih, kot so upori in kondenzatorji. Intenzivnost odpovedi je malce večja tudi zato, ker smo predpostavili, da naprava deluje v zemeljskem, mobilnem okolju ( $G_M$ ). Sicer to ni čisto res, vendar se razvojni sistem zaradi svoje namembnosti veliko prenaša, prav tako pa ni zaščiten pred zunanjimi vplivi (prah, svetloba, ...) in tudi ni v ohišju. Zato se nam je zdelo primerno, da vzamemo za izračun slabše pogoje okolja. Za nekaj elementov nam ni uspelo ugotoviti kaj so, oziroma nam jih ni uspelo uvrstiti v poglavja MIL-HDBK-217F. Tem smo dali intenzivnost odpovedi  $\lambda_p = 1$ , kar tudi vpliva na netočnost rezultata.

## 1.3. Telcordia standard

### 1.3.1. Intenzivnost odpovedovanja (*Failure rate*)

Intenzivnost odpovedovanja oziroma Failure rate je pri standardu Telcordia izražena v številu napak na milijardo ( $10^9$ ) obratovalnih ur. Failure rate v času  $t$ ,  $\lambda(t)$ , je definiran kot pričakovano število napak med časoma  $t$  in  $t+1$ , pri čemer mora veljati pogoj, da se nobena napaka v sistemu ali komponenti ni pojavila do časa  $t$ . Slednji pogoj je pomemben, ker nam omogoči razlikovati med intenzivnostjo odpovedovanja in frekvenco odpovedovanja (failure frequency). Za sistem, ki se pogosto kviri in ima veliko napak je intenzivnost odpovedovanja relevantna samo za prvo napako. Ker intenzivnost odpovedovanja temelji pogoju, da se do časa  $t$  ni pojavila nobena napaka, ji pravimo tudi pogojna intenzivnost odpovedovanja. Pojem »frekvenca odpovedi« se uporablja, kadar je mišljena brezpogojna intenzivnost napak na enoto časa.

#### *Izboljšave napovedi zanesljivosti*

Telcordia ponuja tri različne metode izračunov. Spodnja tabela določa terminologijo, uporabljeno v dokumentih Telcordie.

|                                       |  |
|---------------------------------------|--|
| Naprava                               | Električni del z natančno opredeljenimi električnimi značilnostmi. Naprave vključujejo integrirana vezja, diode, upore, itd. |
| Enota                                 | Sklop naprav, običajno na najnižji zamenljivi stopnji. Enote vključujejo vezja, module, napajalnike, itd.                    |
| Sistem                                | Celoten sklop, ki opravlja operativne funkcije.  |
| Steady – state stopnja pokvarljivosti | Konstantna stopnja napak, ki zagotavljajo informacije o dolgoročni učinkovitosti izdelka.                                    |
| Burn-in                               | Obratovanje naprave pod visoko temperaturo in drugimi oteženimi pogoji, z namenom stabilizirati performanse.                 |

Izbira posamezne metode izračuna je odvisna od naših zahtev in količine podatkov, ki so na voljo. V nadaljevanju so navedeni splošni opisi za vsako od treh metod.

#### *Metoda 1 (Black Box):*

|   |   |
|---|---|
| 1 | Napovedi temeljijo na Black box možnosti z enotnim/sistemskih burn-in $\leq 1$ ura. Za vsako napravo se predpostavi, da deluje pri 40 stopinjah Celzija in 50 odstotno nazivno električno napetostjo. |
| 2 | Napovedi temeljijo na Black box možnosti z enotnim/sistemskim burn-in $\geq 1$ ura. Za vsako napravo se predpostavi, da deluje pri 40 stopinjah Celzija in 50 odstotno nazivno električno napetostjo. |
| 3 | To je splošni primer za napovedi zanesljivosti za metodo 1. Lahko se upošteva burn-in naprave ter spreminjajoča temperatura in oteženi pogoji.  |

*Metoda 2 (Black Box z integriranimi laboratorijskimi podatki):*

Namen metode 2 je prilagoditev napovedane MTBF (mean time between failures – čas med napakama) enote ali naprave na temeljih razpoložljivih laboratorijskih oziroma testnih podatkov.

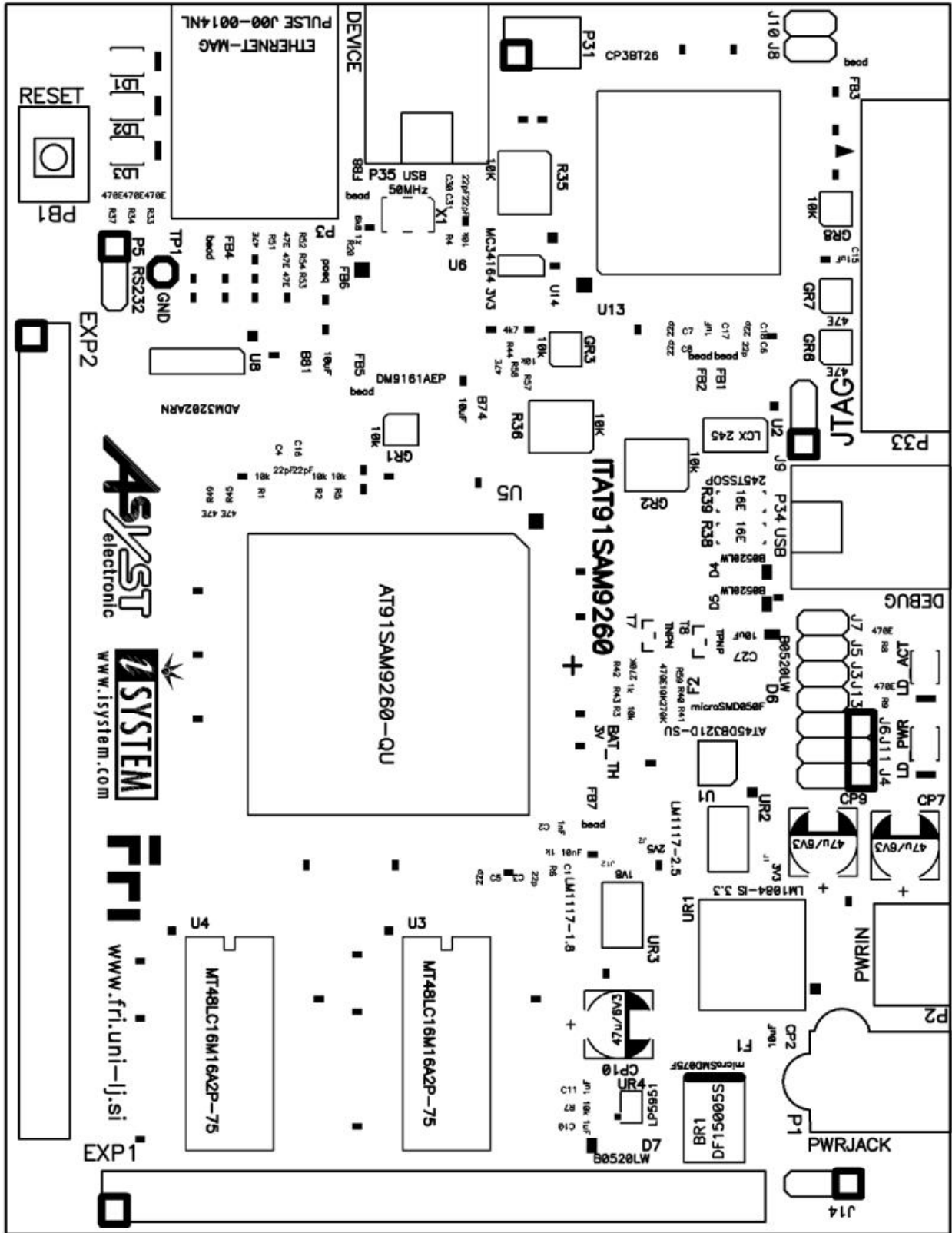
|    |  |
|----|--|
| L1 | Naprave so laboratorijsko testirane in brez burn-in.                         |
| L2 | Enote so laboratorijsko testirane in brez enotske/sistemske burn-in.         |
| L3 | Naprave so laboratorijsko testirane z upoštevanim burn-in.                   |
| L4 | Enote so laboratorijsko testirane z upoštevanim enotskim/sistemskim burn-in. |

*Metoda 3 (Black Box z integriranimi podatki iz terena)*

Namen metode 3 je prilagoditev napovedane MTBF (mean time between failures – čas med napakama) enote ali naprave na temeljih razpoložljivih podatkov iz terena. Metoda 3 je izračunana kot uteženo povprečje opazovanega števila okvar na terenu ter generičnega števila okvar metode 1. Število obratovalnih ur testiranja na terenu določa uteži.

|         |  |
|---------|--|
| III (a) | Zagotavlja predikcijo stopnje napak za naprave, enote ali pod sisteme na temeljih dejanskih in-service performans.   |
| III (b) | Zagotavlja predikcijo stopnje napak za naprave, enote ali pod sisteme na temeljih dejanskih in-service performans kot del drugega sistema. Za te ocene se določijo prilagoditve, glede na to, kakšni so pogoji obratovanja.                          |
| III (c) | Zagotavlja predikcijo stopnje napak za naprave, enote ali pod sisteme na temeljih dejanskih in-service performans podobne opreme istega proizvajalca. Za te ocene se določijo prilagoditve, glede na to, kakšni so pogoji obratovanja obeh sistemov. |

1.3.2. ARM9 razvojni sistem FRI-SMS



### *Intenzivnost odpovedovanja pri FRI-SMS:*

Telcordia standard smo v praksi uporabili za analizo razvojne plošče FRI-SMS z AT91SAM9260 mikroprocesorjem. Pri delu smo uporabljali orodje Relx Architect 2008. Upoštevali smo lastnosti 28 različnih elementov, ki sestavljajo analiziran sistem. Podatki so bili dobljeni iz uradnih specifikacij posameznih elementov. Intenzivnosti odpovedovanja so izračunane z uporabo Metode 1 po Telcordia Issue 2 standardu. Povsod je predpostavljeno delovanje na tleh v kontroliranem okolju in da se vsi elementi uporabljajo ves čas(100% Duty Cycle).

Prva tabela predstavlja podatke dobljene z uporabo Metode 1, Primer 1. Ta primer predpostavlja, da podatki o temperaturi in obremenitvi niso znani, zato predpostavi temperaturo 40°C in 50% obremenitev.

V tem primeru dobimo izračunano intenzivnost odpovedovanja 7.177804 na 10<sup>6</sup> ur obratovanja.

Tabela 1:

| <b>Part Num</b> | <b>Name</b>    | <b>Description</b>  | <b>FailureRate</b> | <b>Quantity</b> |
|-----------------|----------------|---|--------------------|-----------------|
| 1               | System         |   | <b>7,177804</b>    | 1               |
| 2               | LM1084-IS 3.3  | 5A Low Dropout Positive Regulator   | 0,049005           | 1               |
| 3               | LM1117-1.8     | 800mA Low-Dropout Linear Regulator  | 0,042085           | 2               |
| 4               | LM1117-2.5     | 800mA Low-Dropout Linear Regulator  | 0,024503           | 1               |
| 5               | CP3BT26        | Reprogrammable Connectivity Processor with Bluetooth, USB, and CAN Interfaces | 0,556771           | 1               |
| 6               | ADM3202ARN     | Low Power RS-232 Line Driver/Receiver   | 0,020472           | 1               |
| 7               | MC34164        | Micropower Undervoltage Sensing Circuit                                       | 0,030025           | 1               |
| 8               | DM9161AEP      | 10/100 Mbps Fast Ethernet Physical Layer Single Chip Transceiver              | 0,030025           | 1               |
| 9               | AT91SAM9260    | AT91 ARM Thumb Microcontroller  | 0,556771           | 1               |
| 10              | MT48LC16M16A2P | Synchronous DRAM  | 0,191064           | 2               |
| 11              | AT45DB321D-SU  | 32-Megabit 2.7-Volt DataFlash   | 0,363427           | 1               |
| 12              | B0520LW        | 0.5A Surface mount schottky barrier rectifier                                 | 0,089997           | 4               |
| 13              | LCX245         | Low Voltage Bidirectional Transceiver   | 0,020472           | 1               |
| 14              | LP5951         | Micropower, 150mA Low-Dropout CMOS Voltage Regulator                          | 0,020472           | 1               |

|    |            |  |          |    |
|----|------------|--|----------|----|
| 15 | DF15005S   | Miniature Glass Passivated Single-Phase Surface Mount Bridge Rectifier | 0,00235  | 1  |
| 16 | J00-0014NL | 10/100 Base-TX RJ45 1x1 Tab-DOWN 8-pin Integrated Magnetics Connector, | 0,001304 | 1  |
| 17 | 10K        | Resistor   | 0,047156 | 14 |
| 18 | 16R0       | Resistor   | 0,007797 | 2  |
| 19 | 47E        | Resistor   | 0,031    | 9  |
| 20 | 470E       | Resistor   | 0,021236 | 6  |
| 21 | 4K7        | Resistor   | 0,004255 | 1  |
| 22 | 270K       | Resistor   | 0,007797 | 2  |
| 23 | 1K         | Resistor   | 0,007797 | 2  |
| 24 | 6K8        | Resistor   | 0,004255 | 1  |
| 25 | 10uF       | Capacitor  | 4,832812 | 33 |
| 26 | 10nF       | Capacitor  | 0,123111 | 1  |
| 27 | 1uF        | Capacitor  | 0,372015 | 4  |
| 28 | 22p        | Capacitor  | 0,817859 | 10 |
| 29 | 1nF        | Capacitor  | 0,123111 | 1  |

Druga tabela predstavlja podatke dobljene z uporabo Metode 1, Primer 3. Pri tem primeru se vzamejo podatki o temperaturi in obremenitvi kot so navedene za vsak element posebej.

Dobimo izračunano intenzivnost odpovedovanja 4,360074 na  $10^6$  ur obratovanja, kar je precej bolje od prejšnjega primera.

Tabela 2:

| Part Num | Name          | Description                        | FailureRate     | Quantity |
|----------|---------------|------------------------------------|-----------------|----------|
| 1        | System        |                                    | <b>4,360074</b> | 1        |
| 2        | LM1084-IS 3.3 | 5A Low Dropout Positive Regulator  | 0,024703        | 1        |
| 3        | LM1117-1.8    | 800mA Low-Dropout Linear Regulator | 0,021214        | 2        |
| 4        | LM1117-2.5    | 800mA Low-Dropout Linear Regulator | 0,012351        | 1        |

|    |                |   |          |    |
|----|----------------|---|----------|----|
| 5  | CP3BT26        | Reprogrammable Connectivity Processor with Bluetooth, USB, and CAN Interfaces | 0,321084 | 1  |
| 6  | ADM3202ARN     | Low Power RS-232 Line Driver/Receiver   | 0,011806 | 1  |
| 7  | MC34164        | Micropower Undervoltage Sensing Circuit                                       | 0,019568 | 1  |
| 8  | DM9161AEP      | 10/100 Mbps Fast Ethernet Physical Layer Single Chip Transceiver              | 0,019568 | 1  |
| 9  | AT91SAM9260    | AT91 ARM Thumb Microcontroller  | 0,321084 | 1  |
| 10 | MT48LC16M16A2P | Synchronous DRAM  | 0,110185 | 2  |
| 11 | AT45DB321D-SU  | 32-Megabit 2.7-Volt DataFlash   | 0,154365 | 1  |
| 12 | B0520LW        | 0.5A Surface mount schottky barrier rectifier                                 | 0,07491  | 4  |
| 13 | LCX245         | Low Voltage Bidirectional Transceiver   | 0,011806 | 1  |
| 14 | LP5951         | Micropower, 150mA Low-Dropout CMOS Voltage Regulator                          | 0,011806 | 1  |
| 15 | DF15005S       | Miniature Glass Passivated Single-Phase Surface Mount Bridge Rectifier        | 0,001796 | 1  |
| 16 | J00-0014NL     | 10/100 Base-TX RJ45 1x1 Tab-DOWN 8-pin Integrated Magnetics Connector,        | 0,000799 | 1  |
| 17 | 10K            | Resistor  | 0,039251 | 14 |
| 18 | 16R0           | Resistor  | 0,00649  | 2  |
| 19 | 47E            | Resistor  | 0,025803 | 9  |
| 20 | 470E           | Resistor  | 0,017676 | 6  |
| 21 | 4K7            | Resistor  | 0,003542 | 1  |
| 22 | 270K           | Resistor  | 0,00649  | 2  |
| 23 | 1K             | Resistor  | 0,00649  | 2  |
| 24 | 6K8            | Resistor  | 0,003542 | 1  |
| 25 | 10uF           | Capacitor   | 2,962813 | 33 |
| 26 | 10nF           | Capacitor   | 0,075474 | 1  |
| 27 | 1uF            | Capacitor   | 0,228068 | 4  |
| 28 | 22p            | Capacitor   | 0,501398 | 10 |
| 29 | 1nF            | Capacitor   | 0,075474 | 1  |



### 1.3.3. Zaključek

Do nekaterih podatkov posameznih komponent, ki so potrebni za določitev intenzivnosti odpovedovanja po standardu Telcordia nismo uspeli priti, zato je lahko rezultat dokaj drugačen od tistih »točnih«. Kljub temu pa smo vpisali v orodje Relex tiste najpomembnejše podatke, ki jih zahteva omenjeni standard npr. temperaturno območje delovanja komponente, vrednosti upora, kondenzatorjev itd.

Analiziranje z orodjem Relex Architect 2008 bi bilo precej lažje, če bi imeli na voljo polno verzijo, ki omogoča pregled podatkov za 420.000 različnih komponent, in ne demo verzije.

## 1.4. Komentar k zanesljivosti strojne opreme

Vidimo, da pri izračunu po različnih standardih pridemo do različnih rezultatov. Po MIL analizi je odpoved 783.8679004 Odpovedi/10<sup>6</sup> ur, pri Telcordii pa 7.177804, oziroma 4,360074 odpovedi/10<sup>6</sup> ur obratovanja, odvisno od tega po kateri metodi delamo.

Največji razlog za tako razliko je starost obeh standardov. Z orodjem Relex lažje najti točno določeno komponento vezja, saj je okolje mlajše in vsebuje več novejših komponent in njihovih podatkov o zanesljivosti, ki pa jih MIL nima (in jih leta 1991 verjetno še ni predvideval). Pri MIL se je zato velikokrat upoštevala najslabša možna vrednost, saj prave ni bilo moč najti, kar pri količini parametrov, ki se jih uporablja za izračun zanesljivosti elementa, pomeni kar visoko številko.

Razliko med standardoma smo seveda pričakovali, nismo pa si mislili, da bo tako velika. Kljub temu smo zadovoljni z delom. Če bi danes razvijali kakšen sistem bi za izračun zanesljivosti uporabili novejšo orodje (torej standard Telcordia). Vseeno pa bi se morda oprli na MIL. Čeprav številke ne izražajo realnih vrednosti, se da iz njih razbrati kateri elementi so najbolj kritični v sistemih, kar je zelo dobro vedeti pri razvoju sistemov.

## 2. Zanesljivost programske opreme

### 2.1. Pomen analize zanesljivosti programske opreme

Programska oprema v kontekstu zanesljivosti za razliko od strojne opreme prestaja le dve fazi, *testno* in *eksploatacijsko*. Da omogočimo modeliranje, običajno predpostavimo:

- število odpovedi med delovanjem in intenzivnost odkrivanja napak med testiranjem sta proporcionalni številu napak v kodi,
- napaka je takoj po odkritju v trenutku odpravljena,
- v sistemu ne nastajajo nove napake (npr. ob odpravi obstoječih napak ali ob uvajanju novih funkcij).

Število napak (in s tem odpovedi) s časom testiranja pada, v neki točki pa proizvajalec produkt pošlje na tržišče. Odprava napake, ko je produkt že na tržišču, je mnogo težja in zato dražja, zato se proizvajalci zanašajo na modele zanesljivosti programske opreme, da ugotovijo število napak, ki še ostajajo v sistemu, in trend tega števila v primeru, da se odločijo za nadaljnje testiranje.

### 2.2. Programska oprema ARM modula

Na pomnilniških karticah vezja sta naložena operacijski sistem Debian GNU/Linux 5.0 in zagonski nalagalnik. Predpostavimo, da je ta programska oprema popolnoma zanesljiva.

V tem poglavju predvidevamo uporabo ARM modula za pomožni spletni strežnik (npr. da v primeru odpovedi glavnega spletnega strežnika obiskovalce obvesti o začasni nedosegljivosti spletne strani). Na podlagi javno dostopnih podatkov o odpovedih bomo analizirali zanesljivost spletnega strežnika (tj. programske opreme) Apache HTTP Server.  $\% \text{Tu je treba nabruhat odstavek ali dva o tem, kako je ta zadeva kul (tudi ker pokuri malo štroma) za kakšne enostavne (ne preveč obiskane) webserverje. Zato smo torej naložili gor Apache HTTP Server in nas zanima, če je sploh zanesljiv (za OS bomo pač to predpostavili ...)}$

### 2.3. Vir podatkov

Podatke o najdenih napakah v Apache HTTP Server smo poiskali v javni bazi "hrošcev" (Bugzilli) na strani: <https://issues.apache.org/bugzilla/>. Omenili smo že, da smo predpostavili, da se vsaka napaka hipoma odpravi. Zato smo vsako prijavo uspešno odpravljenega hrošča zabeležili kot odpoved (kot da se je napaka šele takrat pojavila). Zaradi obsega celotne baze podatkov o napakah smo uporabili samo podatke od leta 2006 naprej.

Očitno je, da so ti podatki iz eksploatacijske dobe, vendar morajo biti tudi napake, najdene v tej dobi, odpravljene. Lahko si torej predstavljamo, da so ti podatki iz tako imenovane faze *beta testiranja*, ko proizvajalec programske opreme da program na voljo izbranim strankam, da le-te pomagajo pri iskanju in odpravljanju nadaljnjih napak. Iskanje napak v programski opremi namreč s časom postaja vse težje, saj so najbolj očitne napake opažene in odpravljene v zgodnejših fazah testiranja.

Tekom let, iz katerih imamo podatke, so bile v Apache HTTP Server-ju uvedene tudi nove funkcije, zato so se pojavile tudi nove napake. To pomeni, da je občasno število napak hipoma naraslo, nato pa znova začelo upadati. Tovrstne anomalije bomo zanemarili, saj jih modeli ne znajo upoštevati.

## 2.4. Uporabljeni modeli

Držali smo se nasveta iz literature [1] ter poskušali podatkom prilagoditi več modelov, da bi ugotovili, kateri se najboljše prilega podatkom. Za primerjavo smo izbrali linearen model, model Jelinski-Moranda, osnovni model Muse, in model Musa-Okumoto. (Zgodovinsko gledano, je bil prvi model za napovedovanje zanesljivosti programske opreme, ki je kot vhodne podatke uporabljal čase med napakami, Jelinski-Moranda.)

Vse modele lahko uporabljamo kot „črne škatle“: podamo jim zahtevane podatke, kot rezultat pa dobimo napoved števila preostalih napak, ter napoved, kdaj bodo te napake najdene in odpravljene (ponovno ob predpostavki, da je napaka hipoma odpravljena).

Kot implementacijo modelov smo uporabili programsko orodje CASRE iz Nasinega *Jet Propulsion Laboratory*-ja, ki je na voljo brezplačno na [http://www.openchannelsoftware.com/projects/CASRE\\_3.0](http://www.openchannelsoftware.com/projects/CASRE_3.0).

Za določitev parametrov modela smo vsakič uporabili minimizacijo srednjekvadratične napake.

### Model Jelinski-Moranda

Je eden izmed prvih in najenostavnejših modelov. Predpostavlja, da ima program na začetku  $\$N\$$  napak, da so vse napake enako verjetne in med seboj neodvisne, da so intervali med napakami med seboj neodvisni, da je vsaka napaka vedno takoj odpravljena in s popravkom v kodo niso vnešene nove napake. Za povezavo med napakami in odpovedmi tudi predpostavlja, da je intenzivnost odpovedovanja programske opreme med napakami konstantna in proporcionalna številu preostalih napak v programu.

### Osnovni model Muse

Tudi ta model predvideva takojšnjo odpravo napak in neodvisnost odpovedi, katerih število je privzeto kot proporcionalno številu napak v kodi programa. Model privzame linearno padajočo funkcijo hazarda.

### Model Musa-Okumoto

Musa-Okumotov model temelji na nehomogeni Poissonovi porazdelitvi (torej spada med NHPP -- *Non-Homogeneous Poisson Process* -- modele). Pričakuje, da število odpovedi (in s tem najdenih napak) skozi čas testiranja eksponentno pada.

## 2.5. Linearni model

Model predpostavlja linearno naraščanje meodpovednih časov v odvisnosti od zaporedne številke odpovedi.

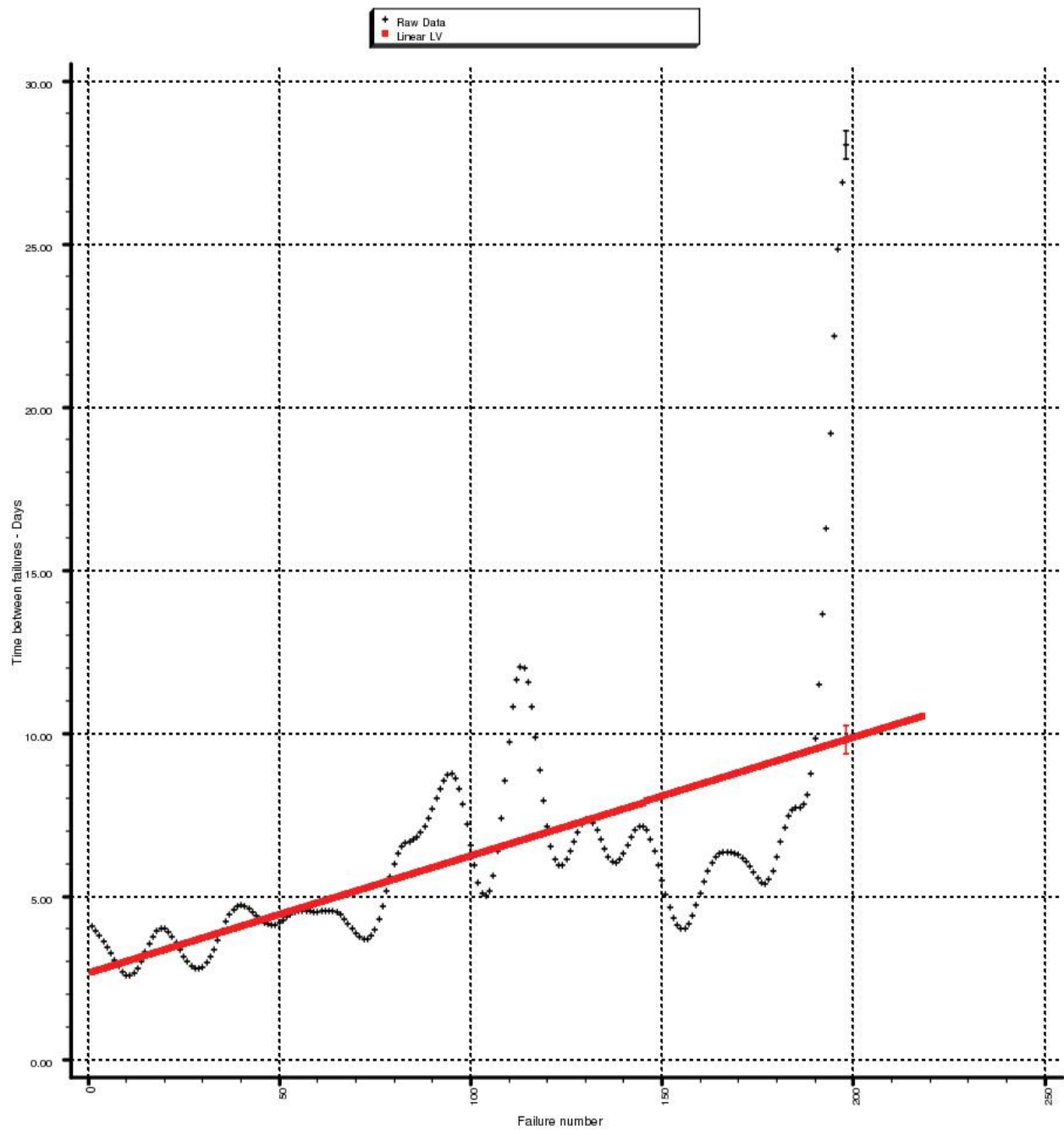
## 2.6. Rezultati

Graf 1 prikazuje medodpovedne čase v odvisnosti od *zaporedne številke odpovedi*: s črno barvo so označeni zabeleženi podatki, z rdečo pa napoved linearnega modela. Opazimo, da se model ne prilega podatkom. Model je sicer enostaven, vendar precej pesimističen in nerealen. Napoveduje počasno linearno rast medodpovednih časov, v podatkih pa opazamo drugačen trend.

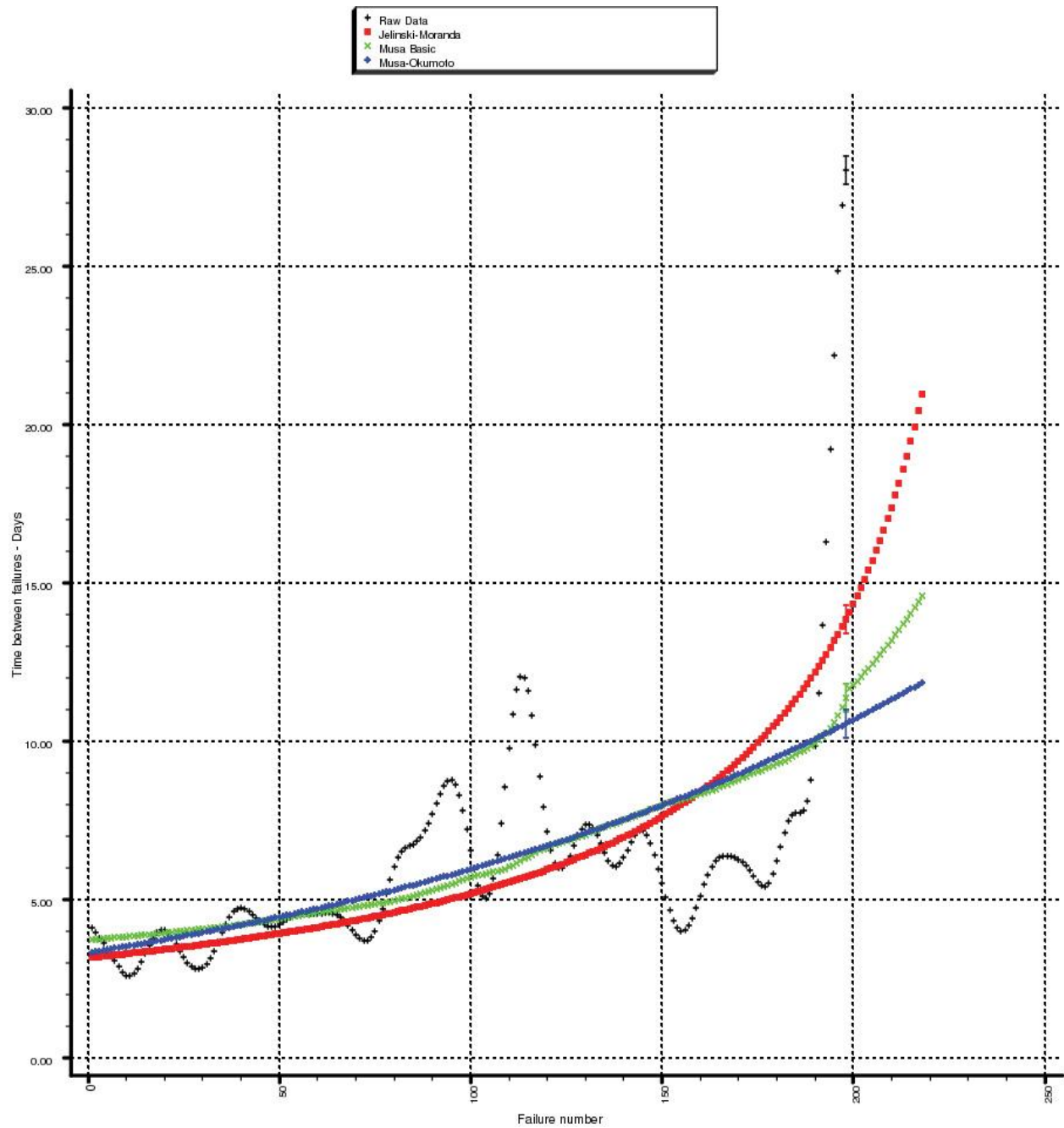
Graf 2 prikazuje iste podatke za tri druge modele: Jelinski-Moranda, osnovni model Muse, in model Musa-Okumoto. Opazimo da se podatkom najbolj prilega model Jelinski-Moranda, sledi osnovni Musa model ter nazadnje še Musa-Okumoto. Na črtah, ki predstavljajo rezultate modelov, je z navpično črto označena točka, od katere naprej nimamo več izmerjenih podatkov; krivulje od tam naprej nakazujejo predikcijo za naslednjih 20 odpovedi.

Graf 3 za iste tri modele prikazuje pričakovano število najdenih (odpravljenih) napak na dan testiranja. Jelinski-Moranda na primer 2500 dni po začetku testiranja napove v povprečju 0.015 odpovedi (oziroma odpravljene napake) na dan (tj. 1 odpoved v 67 dneh). številka je nižja od preostalih dveh, saj model napove več odpravljenih napak v zgodnejši fazi testiranja.

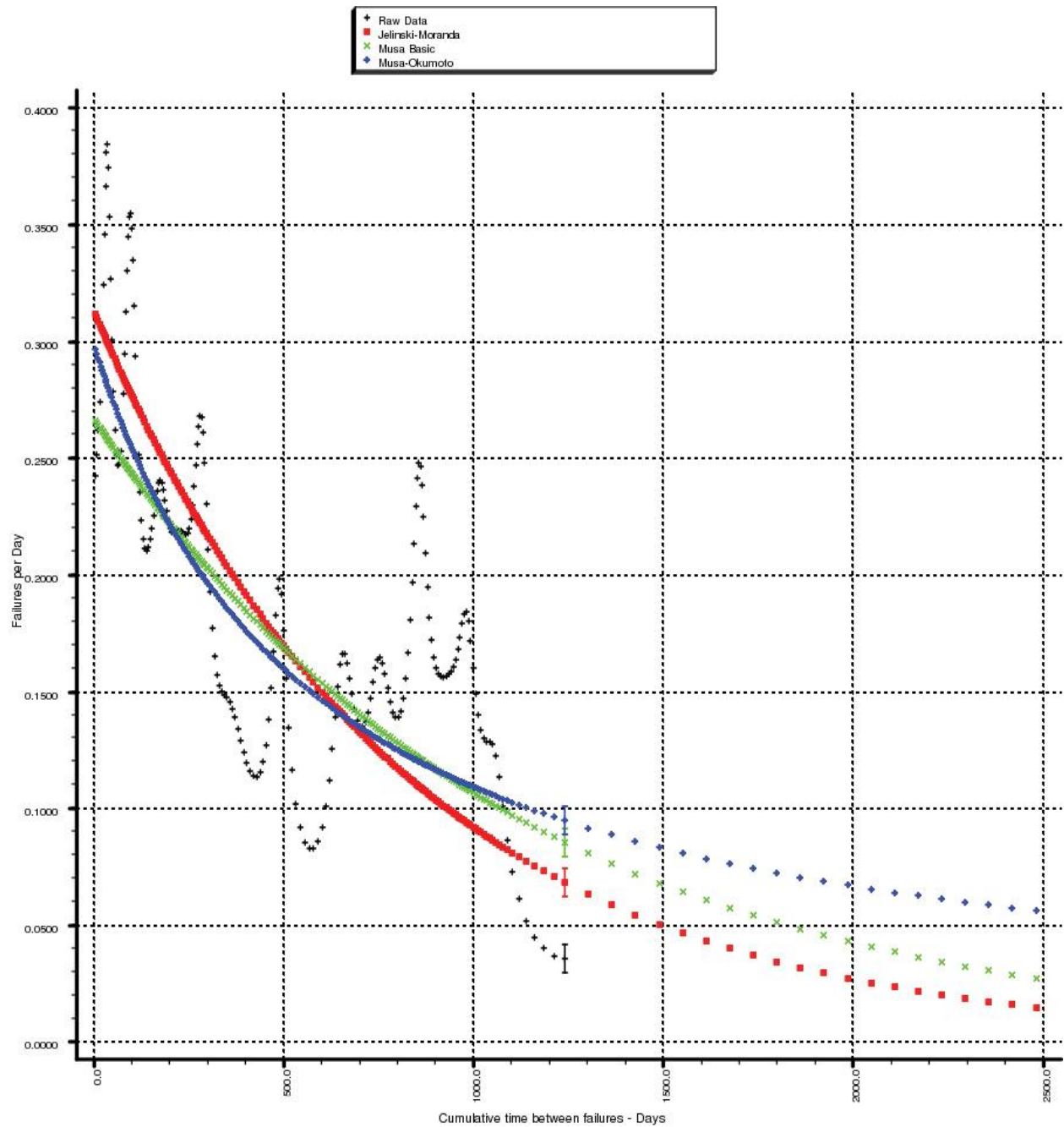
Zadnji graf (4) prikazuje napoved zanesljivosti programske opreme po 28 dneh, torej verjetnost delovanja programske opreme po naslednjih 28 dneh obratovanja, v odvisnosti od časa testiranja. Ponovno opazimo, da si modeli sledijo po istem vrstnem redu. Najvišjo zanesljivost napove Jelinski-Moranda, okoli 0.15 v trenutku, ko smo odpravili zadnjo trenutno znano napako (tj. 1250 dni po začetku testiranja), napove pa zanesljivost 0.7 po 20 nadaljnjih odpravljenih napakah, tj. po 2500 dneh od začetka testiranja.



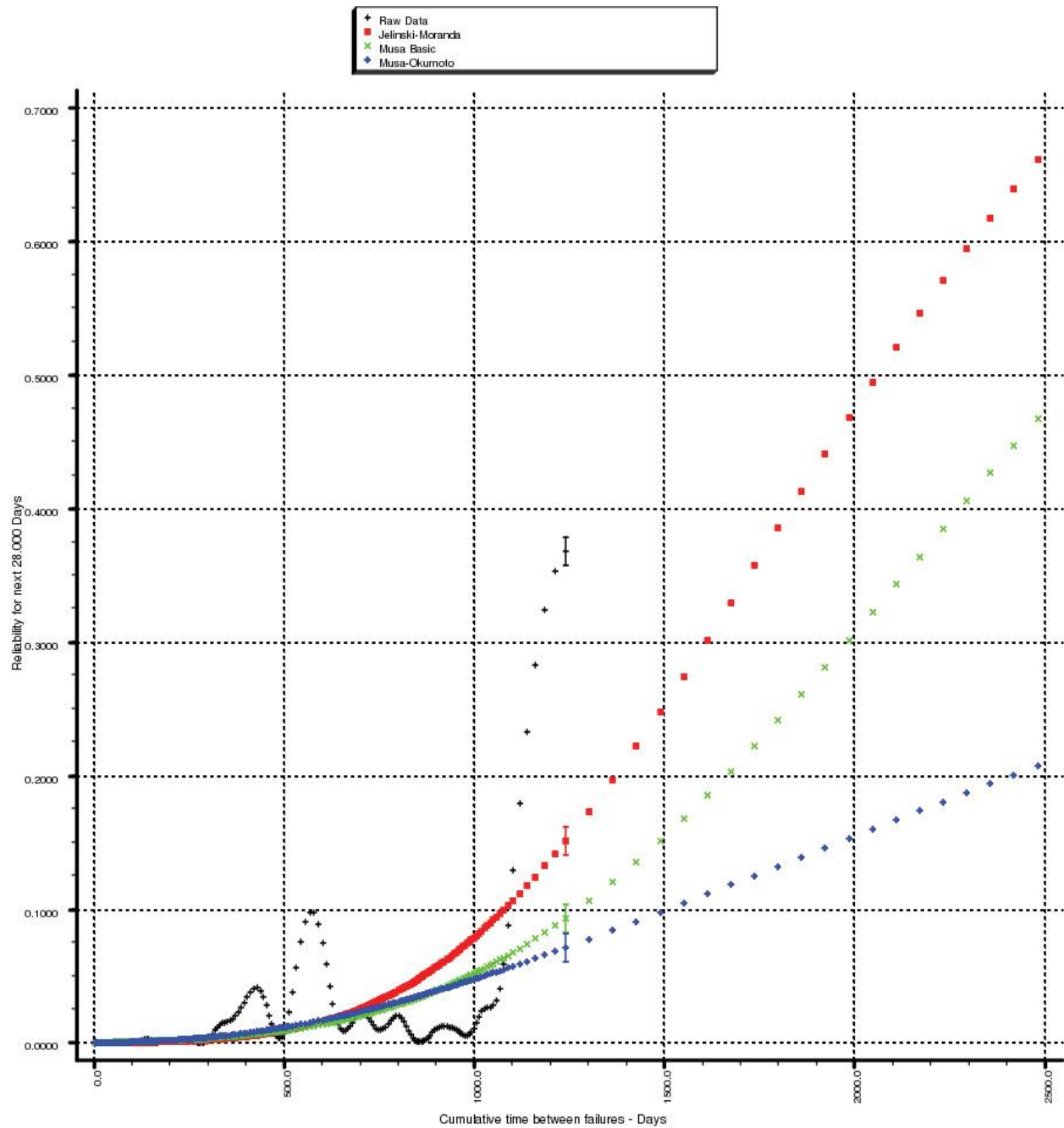
**Graf 1:** Linearen model medodpovednih časov.



**Graf 2:** Medodpovedni časi: primerjava modelov Jelinski-Moranda, osnovnega modela Muse in modela Musa-Okumoto.



**Graf 3:** Napoved pričakovanega števila odpovedi na dan testiranja: model Jelinski-Moranda, Musa-Okumoto in osnovni model Muse.



**Graf 4:** Napoved zanesljivosti v 28 dneh -- tj. verjetnosti, da se bo sistem naslednjih 28 dni deloval brez odpovedi -- v odvisnosti od časa testiranja v dneh.



## 2.7. Ugotovitve

Po primerjavi modelov smo ugotovili, da se model Jelinski-Moranda (med izbranimi modeli) najbolj prilega podatkom.

Ker je namen teh modelov tudi napovedati preostalo število napak, smo za ta model omenjeno število izračunali. Model je pokazal, da je bilo napak na začetku 256, ter da smo jih s testiranjem odpravili 198. Sledi, da jih je v programu ostalo še 58.

Glede na napovedano zanesljivost štiritedenskega delovanja (0.15) se nam zdi smiselno program še naprej testirati pred dokončnim izidom, vendar je to ugotovitev potrebno komentirati. Namestitev programske opreme, katere zanesljivost modeliramo, je po svetu ogromno, odpovedi vseh instanc pa smo obravnavali skupaj. V našem primeru bi vse postavitve sistema tekle po praktično enakih poteh (serviranje ene in iste statične spletne strani), zato jih lahko obravnavamo kot eno samo instanco. Če predpostavimo enakomerno porazdelitev odpovedi preko vseh uporabnikov (instanc), bi morali medodpovedne čase množiti s številom uporabnikov, na ta način pa bi dobili mnogo zadovoljivejšo zanesljivost (blizu 1):

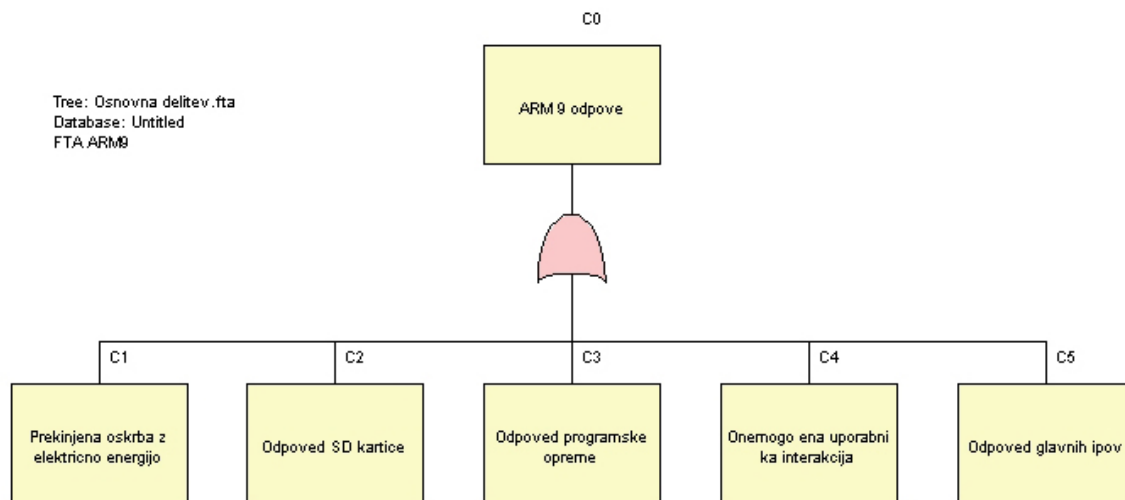
- če kot uporabnike štejemo samo osebe, ki so že kdaj poročale o odpovedi, imamo glede na podatke v Bugzilli opravka z  $u=3528$  uporabniki,
- medodpovedni čas v trenutku, ko smo odpravili zadnjo trenutno znano napako, je po modelu Jelinski-Moranda enak  $MTBF=13.86$  dni,
- naša instanca Apache HTTP Server-ja ima torej  $MTBF' = MTBF \cdot u = 48898.08$  dni,
- zanesljivost po 28 dneh delovanja je potem po modelu Jelinski-Moranda enaka 0.999455.

### 3. Metoda FTA

Fault Tree analiza je ena od najbolj uporabljenih metod v sistemski analizi zanesljivost . To je postopek za določitev različnih kombinacij strojne in programske okvare in človeških napak, ki lahko povzroči pojav določenih nezaželenih dogodkov na ravni sistema. Analiza se začne s splošno sklepom, iz katerega se skuša razviti drevo vzrokov, ki so pripeljali do vzroka tega sklepa. To je pogosto opisano kot "top-down" pristop.

Glavni namen FTA analize je oceniti verjetnost vrhnjega dogodka z uporabo analitične in statistične metode. Ti izračuni vključujejo sistem količinske zanesljivosti in vzdrževanja podatkov, kot so verjetnost odpovedi, okvare ali popravila. FTA analiza lahko zagotovi koristne podatke o verjetnosti okvare, in sredstva, s katerimi bi takšne okvare se lahko pojavijo.

#### 3.1. Vzroki za odpoved

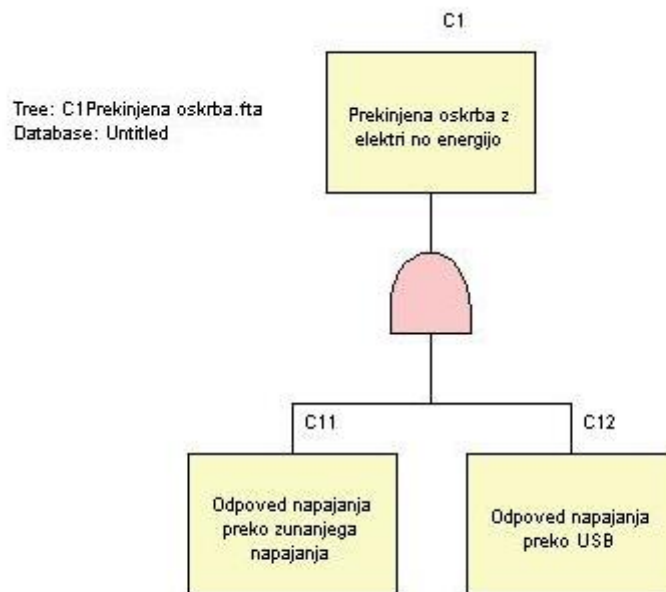


#### *Prekinjena oskrba z električno energijo*

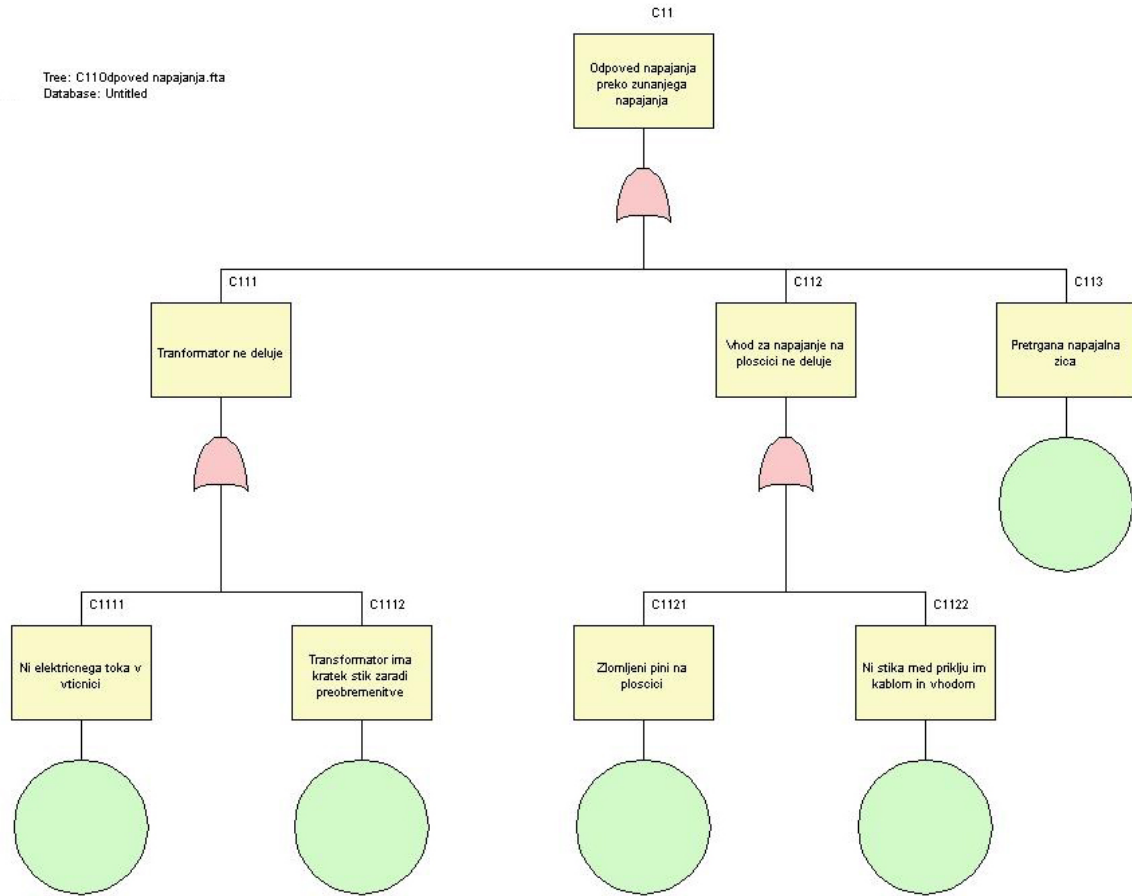
ARM 9 se lahko napaja direktno preko USB priključka ali preko napajanja iz vtičnice preko transformatorja. Lahko je priklopljeno tudi oboje, napajanje iz vtičnice in USB napajanje. Če oboje odpove sistem preide v stanje nedelovanja.

- Odpoved napajanja preko zunanjega napajanja
  - Transformator ne deluje: Če pride do kratkega stika zaradi preobremenitve lahko transformator odpove. To se lahko zgodi na primer če udari strela v električno napeljavo, kamor je priključen transformator. Drugi možen vzrok je, da v transformator ne prihaja tok iz omrežja, recimo pri nevihtah lahko pride do poškodb električnega omrežja v stavbi ali pregori varovalka.

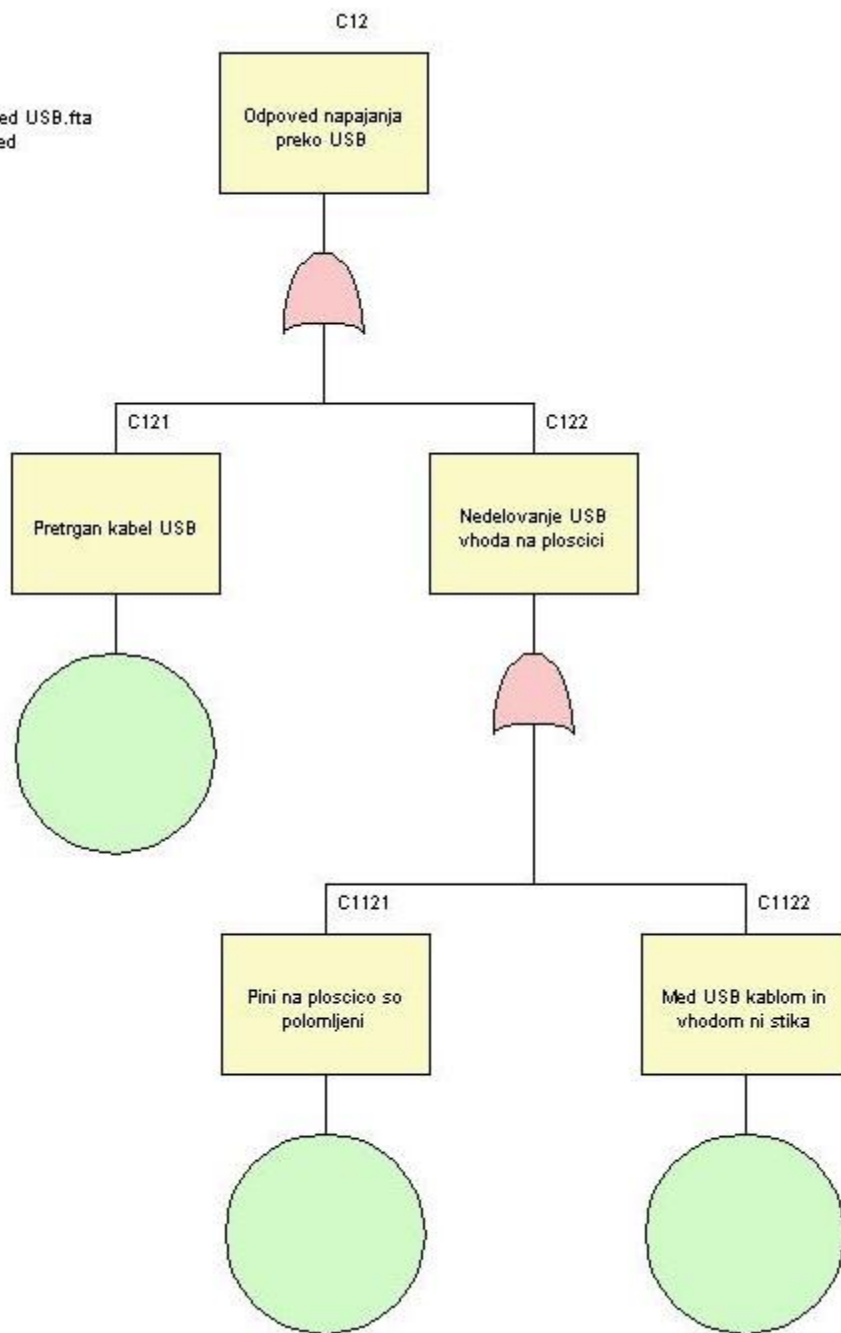
- Vhod za napajanje na ploščici ne deluje: Pri nepravilni uporabi same ploščice lahko pride do tega, da odpove vhod za napajanje ploščice. Vhod je pricinjen na ploščico in lahko se zgodi da se napajalna nožiča zvije in odlomi. Tedaj tok pride do vhodnega priključka a ne more napajati vezja. Če priključek za napajanje velikokrat vtaknemo in iztaknemo lahko pride do tega, da tudi če je napajalni kabel priklapljen v vhod, znotraj vhoda ni stika in tok ne more steči ali slabo teče.
  - Pretrgana napajalna žica: Če pride do pretrganja napajalne žice tok od transformatorja do vhoda za napajanje ne more teč in ploščica ostane brez napajanja.
- Odpoved napajanja preko USB
    - Pretrgan kabel USB: USB kabel je sestavljen iz 2 paric, če pride do pretrganja znotraj para podatkovnih žic bo napajanje še vedno delovalo, če pa se pretrga napajalna parica bo sistem ostal brez napajanja.
    - Nedelovanje USB vhoda na ploščici: Pri dolgotrajni uporabi USB kabla za priklop na ploščico lahko pride do obrabe stika med USB vodom in USB vtičem. Če pride do tega je prehod toka moten ali ga sploh ni. Pri nepravilnem rokovanju se lahko USB vhod odlomi od ploščice ali se odlomijo pini s katerimi je USB pricinjen in ploščica ostane brez napajanja.



Tree: C110dpoved napajanja.fta  
Database: Untitled



Tree: C12Odpoved USB.fta  
Database: Untitled



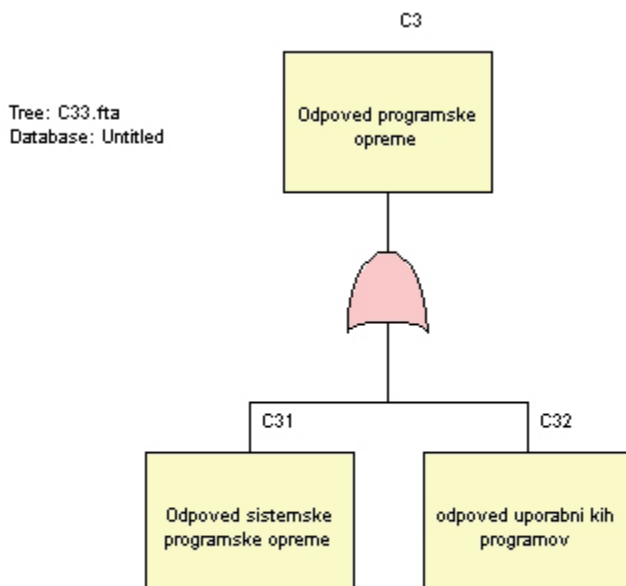
### *Odpoved SD kartice*

ARM ima vgrajeno tudi ležišče za SD kartico, na katero smo v našem primeru naložili operacijski sistem GNU/Linux. V primeru odpovedi SD kartice je sistem neuporaben.

- Odpoved čitalnika kartice: V primeru, da nam odpove čitalec kartic, je to enako kot da bi nam odpovedala SD kartica, saj ne moremo dostopati do podatkov, ki so shranjeni na kartici. Le ta lahko odpove zaradi fizičnih ali pa elektronskih okvar.
- Pokvarjeni podatki na kartici: Na SD kartici je na datotečnem sistemu ext3 posnet del operacijskega sistema (jedro je na flash pomnilniku v glavnem čipu) in uporabniški programi. Pri zapisu na kartico oz. pri branju iz kartice lahko preberemo napačne podatke. Če so podatki napačni v glavnih blokih datotečnega sistema ali v datotekah, ki so potrebne za delovanje sistema bo sistem prenehal delovati.
- Fizična okvara kartice: Kartica odpove tudi v primeru fizične poškodbe. To pomeni, da je na kartici prišlo do neke fizične spremembe, zaradi katere se kartica ne bo obnašala več pravilno, kar bo vodilo v odpoved sistema.

### *Odpoved programske opreme*

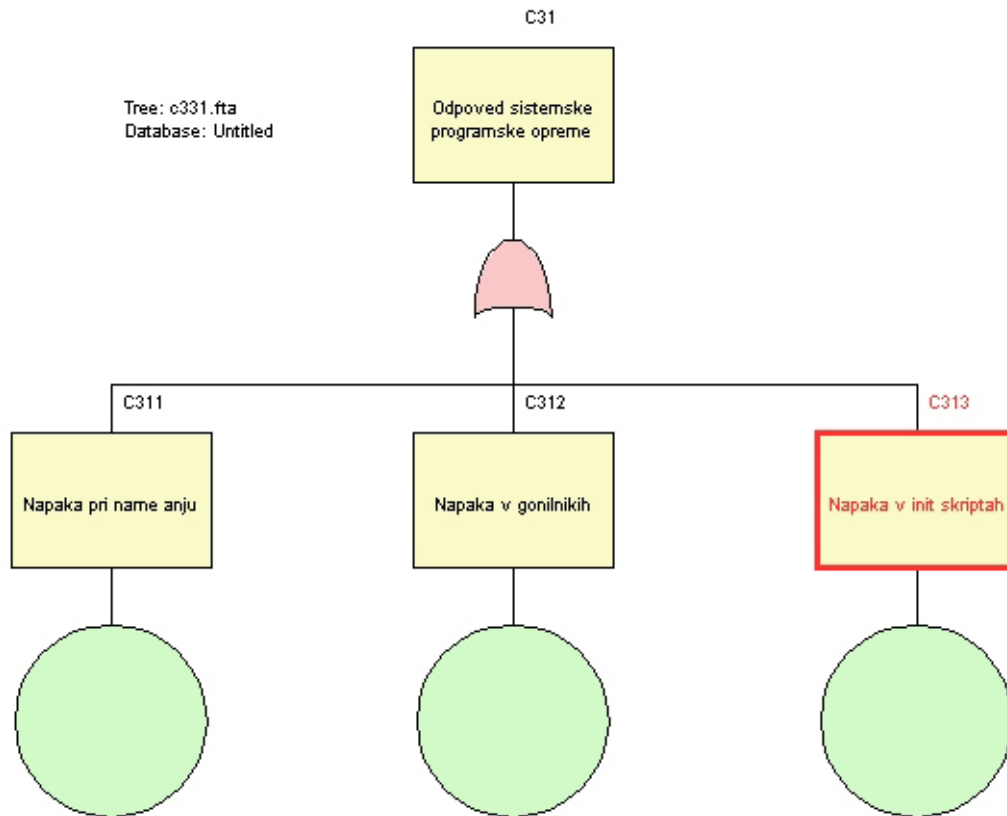
Tudi programska oprema je sestavni del našega sistema. Na razvojni ploščici teče operacijski sistem GNU/Linux, funkcionalnost vršita dva uporabniška programa. Ena je lupinska skripta, ki nabira podatke in jih zapisuje v statičen HTML dokument. Drug uporabniški program pa je spletni strežnik, ki servira to statično stran.



**Odpoved sistemske programske opreme:** Sistemska programska oprema lahko odpove zaradi več vzrokov, ponavadi je kriv uporabnik. Najpogostejši vzrok je prekinjena posodobitev programske opreme.

- Napaka pri nameščanju popravkov: Med nameščanjem popravkov (firmware flash) se prepisuje prejšnji paket programov z novim.
- Prehitro prekinjeno napajanje sistema ali prekinjena mrežna povezava povzroči napako, zaradi katere sistem ne deluje. Do napake lahko pride tudi če odpove uporabnikov računalniški sistem.
- Napaka v gonilnikih: V gonilnikih za vhodno/izhodne naprave, ki so del sistema so lahko programske napake. Napaka se lahko pojavi pri vhodnih (oz. izhodnih) podatkih, ki jih programer ni predvidel. Majhna napaka lahko povzroči upočasnjeno delovanje, napačno delovanje ali odpoved celotnega sistema.
- Napaka v "init" skriptah: Sistem v napravi postavi "init" skripte. Po vrsti nastavljajo:
  1. generator psevdo naključnih števil,
  2. mrežo,
  3. požene se SSH strežnik,
  4. postavi se HTTP strežnik,
  5. zažene se skripta za periodično nabiranje podatkov.

Kritična je odpoved skript 2, 4 in/ali 5. Katera koli od naštetih povzroči nedelovanje sistema. Inicializacijske skripte za delovanje potrebujejo še več drugih sistemskih programov.

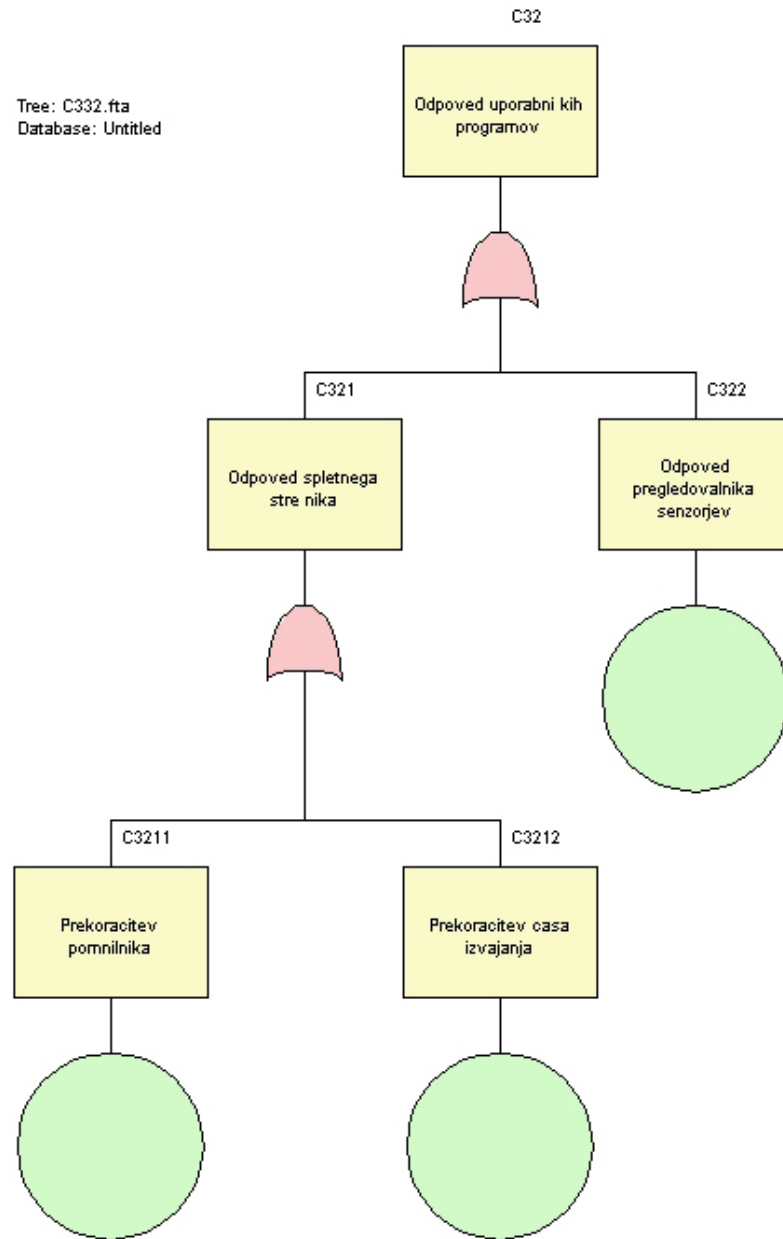


### Odpoved uporabniške programske opreme

- Odpoved spletnega strežnika: Na sistemu teče zelo lahek spletni strežnik. Spletni strežnik lahko odpove zaradi napak v programski kodi in tudi zaradi dejavnosti uporabnika:
  - Prekoračitev pomnilnika: Začasno ali trajno nedelovanje sistema lahko povzroči prekoračitev sistemskega pomnilnika. Do prekoračitve pomnilnika lahko pride zaradi:
    1. napake v programski kodi strežnika (uporabnik pošlje zahtevo, ki ni bila predvidena),
    2. preveč zahtev uporabnikov
  - Prekoračitev časa izvajanja: Če strežnik predolgo obdeluje zahtevo lahko poteče maksimalen čas čakanja odgovora zato uporabnik ne dobi zahtevanih podatkov. Do prekoračitve lahko pride zaradi:
    1. napake v programski kodi strežnika,
    2. preveč zahtev uporabnikov
- Odpoved skripte za pregledovanje senzorjev: To je program, ki pregleda stanja vseh senzorjev in naredi HTML stran s podatki. Nedelovanje programa lahko povzroči uporaba nepodprtih senzorjev, podatki na senzorju v nepredvidenem območju, bolj napaka v programu.



Tree: C332.fta  
Database: Untitled



### *Onemogočena uporabniška interakcija*

O onemogočeni uporabniški interakciji govorimo takrat, ko ne moremo uspešno komunicirati z našim sistemom, ki teče na ARMu. To pomeni, da je dejansko onemogočena uporaba sistema, saj ne moremo izvrševati ukazov oz. ne moremo spremljati rezultatov izvršenih akcij.

- **Onemogočena uporabniška interakcija**

Neodzivanje sistema: Sistem se v tem primeru ne odziva - zmrzne. Uporaba sistema je onemogočena. Do te napake lahko pride zaradi stradanja procesov, ker nek proces "kuri" ves CPU in nam na ta način prekomerno upočasni sistem. Ponavadi je to "leak" v kodi. Do neodzivanja sistema pa lahko pride tudi zaradi napak, ki sistem "sesujejo".

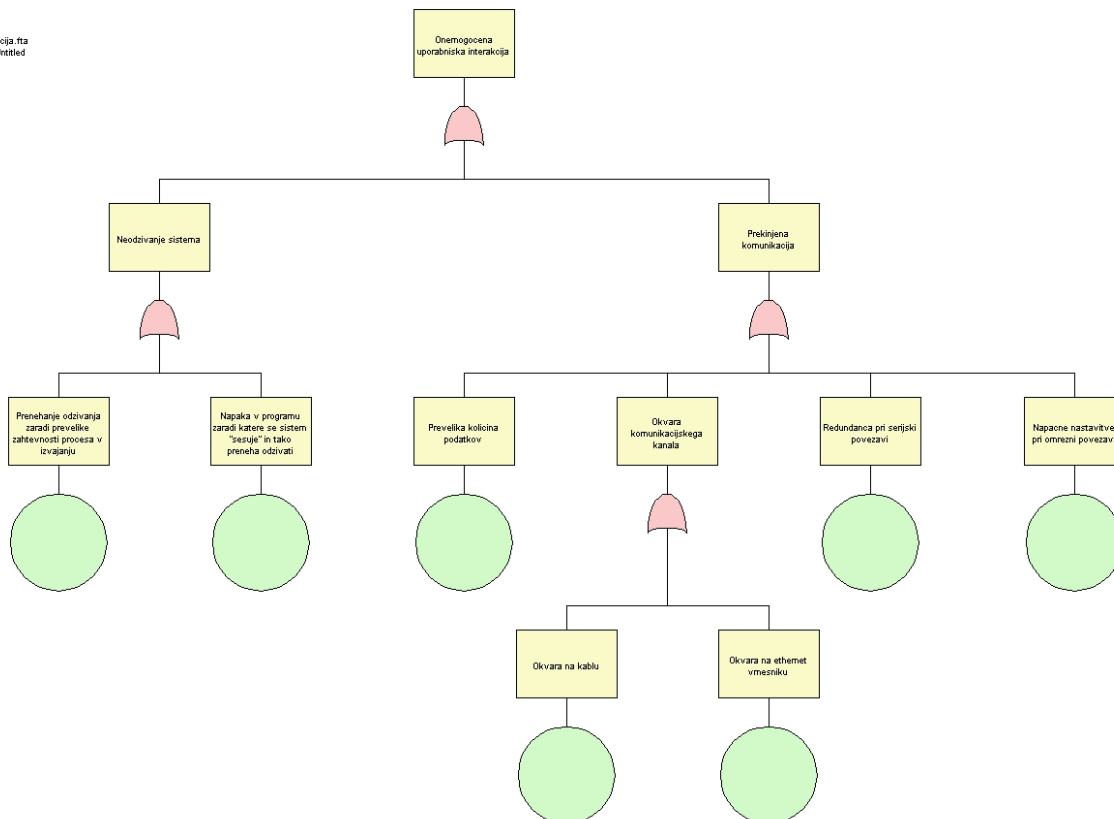
- **Prekinjena komunikacija**

V primeru, da ne uspemo komunicirati z našim sistemom je to ravno tako napaka, zaradi katere je onemogočena uporabniška interakcija s sistemom. V tem primeru ne moremo dostopati do ukazne lupine in tako ne moremo izvrševati ukazov oz. spremljati rezultatov izvrševanja ukazov.

Komunikacija poteka preko ethernet povezave. V primeru, da prenašamo preveliko količino podatkov, sistem ne zmore posredovati tolikšne količine in lahko tako postane takšna povezava neodzivna. Spet pa moramo paziti, da imamo pravilne nastavitve omrežne povezave med računalnikom in ARMom, saj sicer komunikacija ne bo mogoča.

Ravno tako predstavlja enako napako okvara na kablu preko katerega smo povezani na ethernet vmesnik ploščice na kateri teče naš sistem.

V primeru, ko bi uporabljali serijsko povezavo, bi težavo predstavljala redundanca.



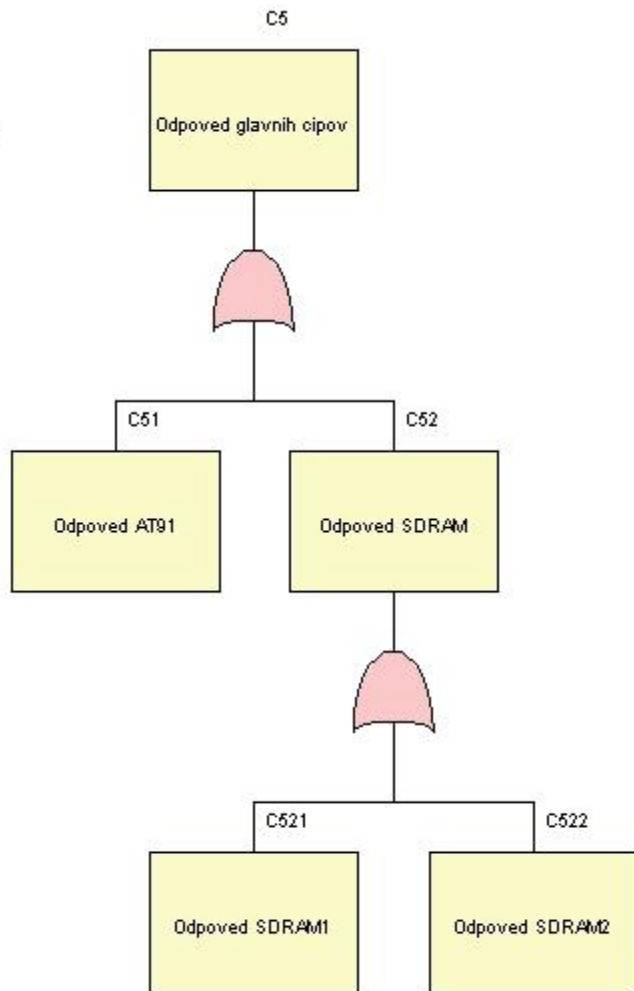
### ***Odpoved glavnih čipov***

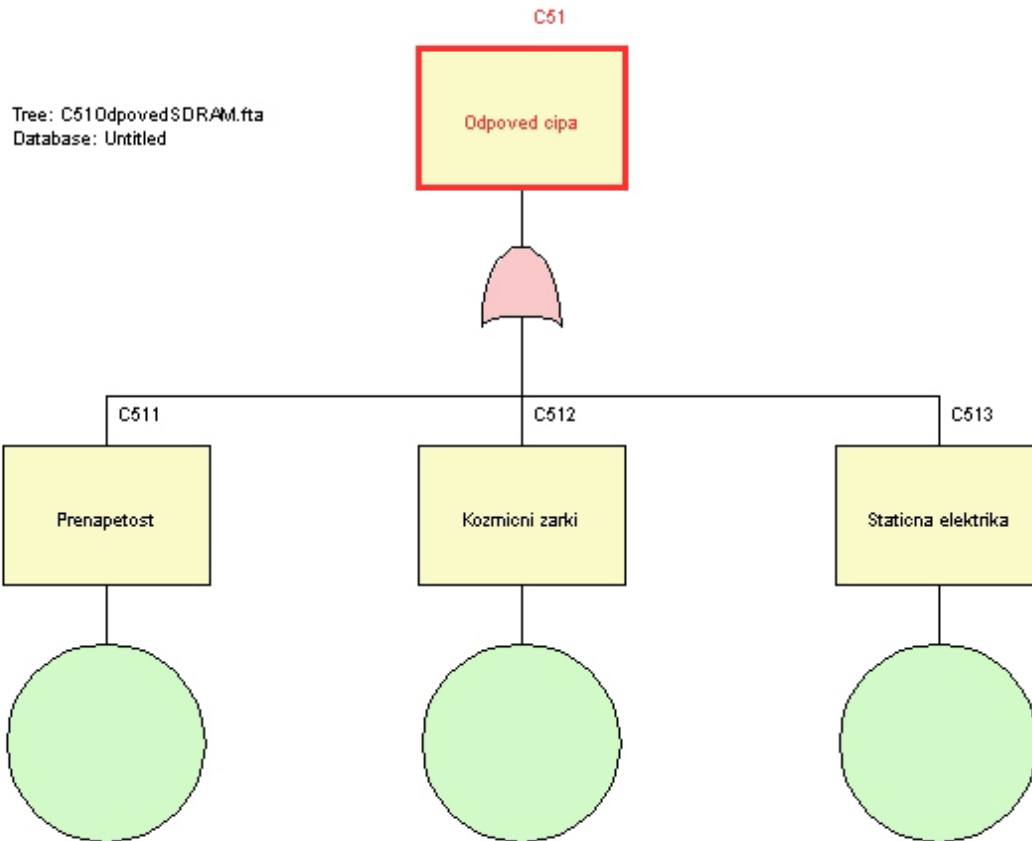
Znotraj glavnega čipa AT91 ARM je procesna enota, vhodno/izhodne naprave, 32KB ROM in dva 4KB SRAMa. Na ARM 9 sta povezana še dva ločena 256 Mbit sinhrona DRAMa. Sam čip pa podpira še zunanje pomnilniške enote.

AT91SAM9260 se nahaja v 208 pinskem PQPF pakiranem čipu, SDRAM pa na ploščici pakiran v po dva 54 pin TSOPII čipa. Za pravilno delovanje sistem je potrebno delovanje vseh pomnilnikov, procesne enote in V/I naprav. Oba čipa odpovedujeta zaradi podobnih razlogov:

- Prenapetost: Lahko pride do prenapetostnega šoka in elektronske komponente zaradi pregrevanja odpovejo. Vzrok prenapetosti je lahko kratek stik ali prenapetost na viru napajanja vezja.
- Kozmični žarki: Če prileti v vezje kozmični žarek iz vesolja, pride do spremembe stanja pomnilnih lokacij znotraj čipa. To privede do logične napake in posledično do nepravilnega delovanja naprave.
- Statična elektrika: Podobno kot pri prenapetosti tudi pri statični elektriki pride do napetostnega šoka, kar privede do odpovedi komponent zaradi pregrevanja. Statično elektriko na čipu lahko povzroči že uporabnik z dotikanjem vezja.

Tree: C55.fta  
Database: Untitled





### 3.2. Interpretacija rezultatov

Glavna pomanjkljivost naše analize je v tem da nimamo definiranih verjetnosti odpovedovanja, torej tudi ne moramo določiti končne verjetnosti odpovedi. Tako smo lahko le predvideli možne okvare in odpovedi, brez točno definiranih verjetnosti.

Obravnani sistem ni namenjen kritičnim aplikacijam. Iz analize je to razvidno saj je redundanca sistema minimalna, obstaja tudi veliko paralelnih možnosti za odpoved.

## 4. FMEA

### 4.1. Opis

FMEA (Failure Mode and Effects Analysis) je preprosto orodje za identifikacijo in klasifikacijo potencialnih načinov odpovedi. Načini odpovedi so vse napake vsistemu, produktu ali procesu, ki lahko nastanejo v procesu načrtovanja ali v procesu sestave. Prav tako razkrije kritične dele sistema, ki potrebujejo posebne operacije za preprečevanje teh odpovedi.

### 4.2. Zgodovina

Zgodovina FMEA je zelo dolga. Še preden je bila razvita kakršnakoli dokumentacija (formalni opis) metode, so izumitelji in strokovnjaki poskušali predvidevati, na kakšen način se lahko izdelek pokvari, še preden je razvit. Razlog za to je draga in časovno potratna metoda razvij in testiraj (vsaka različica izdelka ima lahko napake).

FMEA je bila formalno predstavljena koncu 40-ih z vojaškim standardom 1629 in uporabljena predvsem za razvoj vesoljske in raketne tehnologije. Odpravljala je napake v majhnih dela posamezne tehnologije in s tem zmanjševala stroške.

Porast uporabe je prišel v 60' letih z razvojem tehnologije za pristanek na Luni s posadko. Ford je uvedel FMEA, za varnost, izboljšave zasnove in proizvodnje, v poznih 70' po škandalu z Ford Pintom (ob nesreči se je lahko vnel, ker je bila posoda za gorivo pod zadnjim odbijačem).

### 4.3. Razvoj

FMEA se izvaja v štirih fazah:

Korak 1: Ugotovimo vse možne **načine odpovedi** na podlagi funkcij sistema in njihovih posledic. V primeru ocene resnosti ( $S_i$ ) 9 ali 10 je potrebno, v večini primerov, spremeniti načrt sistema ali procesa (odprava tega načina odpovedi). Resnost teh načinov odpovedi merimo po lestvici od 1 do 10.

| Ocena $S_i$ | Resnost   |
|-------------|---|
| 1           | Malo ali ni vpliva  |
| 2           | Izguba funkcionalnosti ali izgleda, ki jih uporabnik ponavadi napake ne opazi in ne zamenja izdelka |
| 3           | Izguba funkcionalnosti ali izgleda, ki jih uporabnik opazi, a ne zamenja izdelka                    |
| 4           | Izguba funkcionalnosti ali izgleda, ki jih uporabnik opazi in zamenja izdelek                       |
| 5           | Napačno delovanje sekundarne funkcije   |
| 6           | Odpoved sekundarne funkcije   |
| 7           | Napačno delovanje primarne funkcije   |
| 8           | Odpoved primarne funkcije funkcij   |
| 9           | Ogrožena je varnost ali odpoved sistema, z opozorilom   |
| 10          | Ogrožena je varnost ali odpoved sistema, brez opozorila   |

Korak 2: Ocenimo **pogostost pojavitve** ugotovljenih načinov odpovedi. Oceno ( $O_i$ ) pogostosti merimo z lestvico od 1 do 10.

| Ocena $O_i$ | Pogostost pojavitve |
|-------------|---------------------|
| 1           | 1 izmed 1,000,000   |
| 2           | 1 izmed 250,000     |
| 3           | 1 izmed 50,000      |
| 4           | 1 izmed 10,000      |
| 5           | 1 izmed 5,000       |
| 6           | 1 izmed 1,000       |
| 7           | 1 izmed 250         |
| 8           | 1 izmed 50          |
| 9           | 1 izmed 10          |
| 10          | 1 izmed 2           |

Korak 3: Ocenimo **težavnost detekcije** ugotovljenih načinov odpovedi. Tu ne le opozorimo na morebitne težave odkrivanja načinov odpovedi, amapak tudi predlagamo, kako bi le te preprečili oziroma zmanjšali pogostost pojavitve. Oceno ( $D_i$ ) težavnosti detekcije merimo po lestvici od 1 do 10.

| Ocena $D_i$ | Pogostost pojavitve                       |
|-------------|---|
| 10          | Skoraj nična verjetnost odkritja napake   |
| 9           | Zelo neverjetno, da odkrijemo napako      |
| 8           | Neverjetno, da odkrijemo napako           |
| 7           | Zelo nizka verjetnost odkritja napake     |
| 6           | Nizka verjetnost odkritja napake          |
| 5           | Srednja verjetnost odkritja napake        |
| 4           | Srednje visoka verjetnost odkritja napake |
| 3           | Visoka verjetnost odkritja napake         |
| 2           | Zelo visoka verjetnost odkritja napake    |
| 1           | Odpoved vedno odkrijemo                   |

Korak 4: Za vsak način odpovedi izračunamo t.i. RPN ( $\text{Risk Priority Number}$ ).

$RPN = S_i * O_i * D_i$  Na podlagi RPN lahko preprosto ugotovimo kje so področja, ki so najbolj kritična. Načini odpovedi z najvišjim RPN naj bi imeli prednost pri odpravljanju.

#### 4.4. Koristi in uporaba

FMEA pomaga pri izboljševanju kvalitete, zanesljivosti in varnosti izdelkov, procesov, naprav ali storitev. Zmanjša lahko čas in stroške razvoja, dokumentira in beleži ukrepe (s tem odpravi tveganje za ponovitev istih napak) in izboljša garancijo (manj vrnjenih izdelkov). FMEA lahko uporabljamo pri:

- **Procesih:** analiza proizvodnih procesov in procesov za sestavljanje
- **Načrtovanju:** analiza izdelkov pred proizvodnjo
- **Konceptih:** analiza sistemov in podsistemov v zgodnjih stopnjah načrtovanja

- **Napravah:** analiza naprav pred nakupom
- **Storitvah:** analiza industrijskih storitev preden so na voljo za nakup

#### 4.5. Analiza sistema ARM

|                |                     |                        |                        |
|----------------|---------------------|------------------------|------------------------|
| Funkcija       | Hranjenje BIOS-a    | Glavni pomnilnik       | Procesno jedro         |
| Način odpovedi | Pokvarjen del ROM-a | Odpoved multipleksorja | Okvara registrov       |
| Posledica      | Nedelovanje vezja   | Nedelovanje vezja      | Nedelovanje vezja      |
| Si             | 8                   | 6                      | 8                      |
| Vzrok          | Fizična poškodba    | Prevelik tok           | Prevelik tok           |
| Oi             | 1                   | 2                      | 1                      |
| Detekcija      |                     |                        |                        |
| Di             | 1                   | 1                      | 7                      |
| Akcije         | Trše ohišje         | Prašni filter          | Prednapetostna zaščita |
| RPNi           | 8                   | 12                     | 24                     |

|                |                       |                          |                         |
|----------------|-----------------------|--------------------------|-------------------------|
| Funkcija       | Hranjenje OS-a        | Krmilnik pomnilnikov     | USB krmilnik            |
| Način odpovedi | Slučajni izbris       | Odpoved                  | Uničenje                |
| Posledica      | Neppravilno delovanje | Nedelovanje vezja        | USB naprave ni na voljo |
| Si             | 8                     | 8                        | 4                       |
| Vzrok          | Prevelik tok          | Programska napaka        | Pregrevanje             |
| Oi             | 3                     | 1                        | 3                       |
| Detekcija      |                       |                          | Senzor                  |
| Di             | 1                     | 2                        | 2                       |
| Akcije         | Prenapetostna zaščita | Bolj podrobno testiranje | Vgraditev hladilnika    |
| RPNi           | 24                    | 8                        | 24                      |

|                |                 |                               |                               |
|----------------|-----------------|-------------------------------|-------------------------------|
| Funkcija       | LAN krmilnik    | Povezava do periferije        | Zunanje krmiljenje procesorja |
| Način odpovedi | Uničenje        | Uničenje                      | Uničenje                      |
| Posledica      | LAN ni na voljo | Več naprav ni na voljo        | Ni prekinitev                 |
| Si             | 4               | 8                             | 3                             |
| Vzrok          | Prašno okolje   | Korozija                      | Prevelik tok                  |
| Oi             | 2               | 1                             | 2                             |
| Detekcija      |                 |                               |                               |
| Di             | 2               | 4                             | 4                             |
| Akcije         | Prašni filter   | Ohišje odporno proti koroziji | Prenapetostna zaščita         |
| RPNi           | 16              | 32                            | 24                            |



|                |                               |                       |                         |
|----------------|-------------------------------|-----------------------|-------------------------|
| Funkcija       | Vodilo do periferije          | Ponovni zagon sistema | Napajanje               |
| Način odpovedi | Korozija vodila               | Nedelovanje           | Izguba stika            |
| Posledica      | Večina naprav ni na voljo     | Ne moremo resetirati  | Nedelovanje vezja       |
| Si             | 8                             | 4                     | 8                       |
| Vzrok          | Korozija                      | Nasilen uporabnik     | Prevelik tok            |
| Oi             | 1                             | 4                     | 4                       |
| Detekcija      |                               |                       |                         |
| Di             | 4                             | 1                     | 1                       |
| Akcije         | Ohišje odporno proti koroziji | Ohišje za reset gumb  | Kvalitetnejši materiali |
| RPNi           | 32                            | 16                    | 32                      |

|                |                                |                           |                            |
|----------------|--------------------------------|---------------------------|----------------------------|
| Funkcija       | LAN vtič                       | USB vtič                  | RS232 vtič                 |
| Način odpovedi | Izguba stika                   | Izguba stika              | Zlomljen pin               |
| Posledica      | LAN-a ni na voljo              | USB naprave niso na voljo | Ni komunikacije prek RS232 |
| Si             | 4                              | 2                         | 2                          |
| Vzrok          | Obraba                         | Nepravilna uporaba        | Nasilen uporabnik          |
| Oi             | 4                              | 4                         | 4                          |
| Detekcija      |                                |                           |                            |
| Di             | 1                              | 2                         | 3                          |
| Akcije         | Uporaba kvalitetnih materialov |                           |                            |
| RPNi           | 16                             | 16                        | 24                         |

|                |  |
|----------------|--|
| Funkcija       | Uravnavanje električnih signalov       |
| Način odpovedi | Odpoved                                |
| Posledica      | Možno nedelovanje vezja ali podfunkcij |
| Si             | 7                                      |
| Vzrok          | Tok, okolje, uporabnik                 |
| Oi             | 1                                      |
| Detekcija      |  |
| Di             | 7                                      |
| Akcije         | Uporaba kvalitetnih materialov         |
| RPNi           | 49                                     |

## 4.6. Zaključek

Identificirali smo nekatere napake, ki se lahko pojavijo pri tej napravi. Ugotovili smo, da je največji potencialni rizik v osnovni elektroniki na vezju, saj so napake lahko katastrofalne, njihovo odkritje pa ni nujno - dovolj je, da se napetost na nekem vodilu zelo redko nezaželeno spremeni, da postane celotno vezje neuporabno. Potrebno je pa povedati, da je RPN tudi pri tej vrsti napake zelo majhno. Naše priporočilo je torej, da bi se sicer vezje lahko dodatno izboljšalo v smislu zanesljivosti z izboljšavo kondenzatorjev, uporov in tuljav na vezju, a da to ni zares potrebno.

## 5. Markovska analiza

### 5.1. Markovska analiza splošno (kaj je, kratka predstavitev)

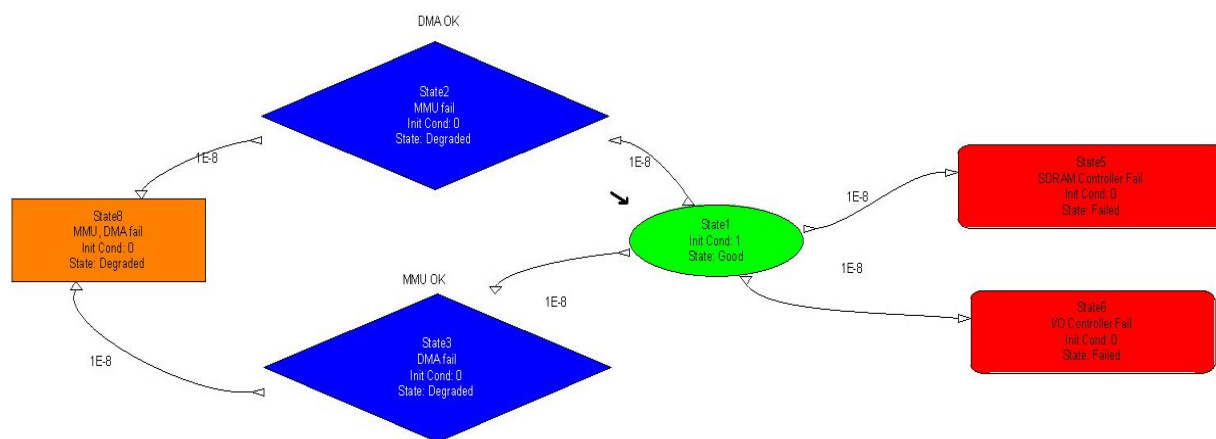
Markovska analiza je metoda, ki uporablja markovske modele kot osnovo za napovedovanje obnašanja sistema. Markovska veriga, oziroma model, kot že vemo, je stohastičen proces, kjer je verjetnost prehoda v naslednje stanje odvisna samo od trenutnega stanja in ne od stanj v preteklosti. Z drugimi besedami, informacija, ki lahko vpliva na prihodno evolucijo sistema, je popolnoma pogojena samo od opisa trenutnega stanja. Torej imamo rojstno-smrtni sistem, kjer je prehod možen le med sosednimi stanji.

### 5.2. Markovska analiza in zanesljivost (prednosti, slabosti...)

Markovska analiza je metoda, ki se uporablja za analizo zanesljivosti in dosegljivosti sistemov, ki so sestavljeni iz med seboj močno povezanih komponent. Osredotoča se na zaporedje dogodkov in analizira težnjo enega dogodka, ki mu sledi naslednji. Z uporabo te analize lahko generiramo novo sekvenco, ki spominja na originalno.

Uporablja se predvsem za modeliranje in natančno analizo manjših sistemov. Zaporedje dogodkov predstavimo z diagramom prehajanja stanj, kjer je vsak dogodek neko deterministično stanje. Glavna pomanjkljivost te metode se kaže pri analizi velikih sistemov, kjer bi bili Markovski diagrami v večini primerov preveliki in prekompleksni za konstrukcijo. Za takšne sisteme se lahko uporablja kombinacija Markovske analize in enostavnejših kvantitativnih modelov. Ker Markovski procesi ne poznajo pomnjenja prejšnjih stanj sistema, so verjetnosti oziroma frekvence prehodov med stanji odvisne le od trenutnega stanja in se s časom ne spreminjajo. Z naraščanjem velikosti diagramov narašča tudi časovna zahtevnost za ročno analizo, zaradi česar se uporabljajo računalniške numerične metode. V našem primeru smo za analizo uporabili programski paket Relex Reliability Studio ([www.relex.com](http://www.relex.com)).

Primer analize odpovedi v računalniškem sistemu:



### 5.3. Markovska analiza sistema z mikrokrmilnikom AT91SAM9620 v RELEX Studio

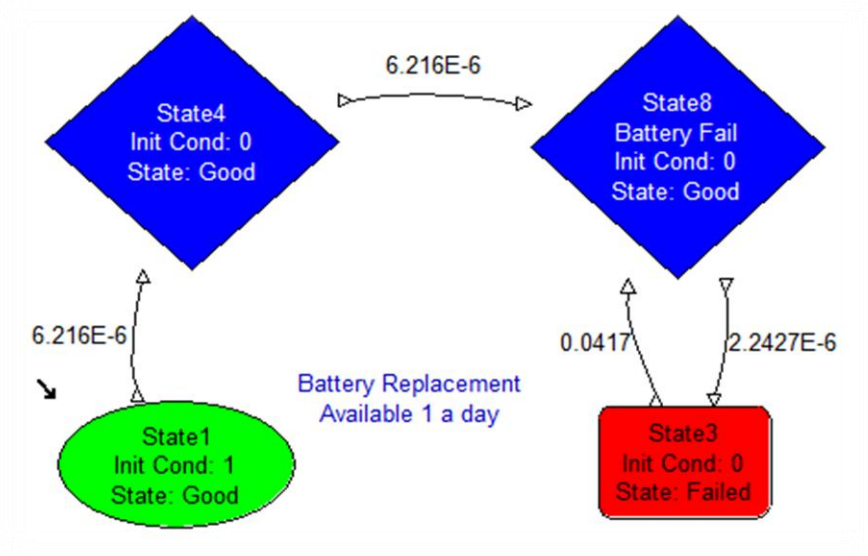
Predpostavimo, da imamo na našem sistemu nameščen FTP strežnik, kar implicira nujno uporabo Ethernet povezave. Imamo še dva trda diska povezana na USB priključkih. Izpad Etherneteta bi v tem primeru povzročil izpad celotnega sistema, kar se zgodi tudi pri izpadu obeh USB priključkov. Sistem lahko napajamo preko dveh priključkov za elektriko ali z baterijo. Analizirali smo tri aspekte, oziroma podsisteme našega sistema:

- analiza napajanja
- analiza pomnilniških komponent
- analiza vhodno izhodnih komponent

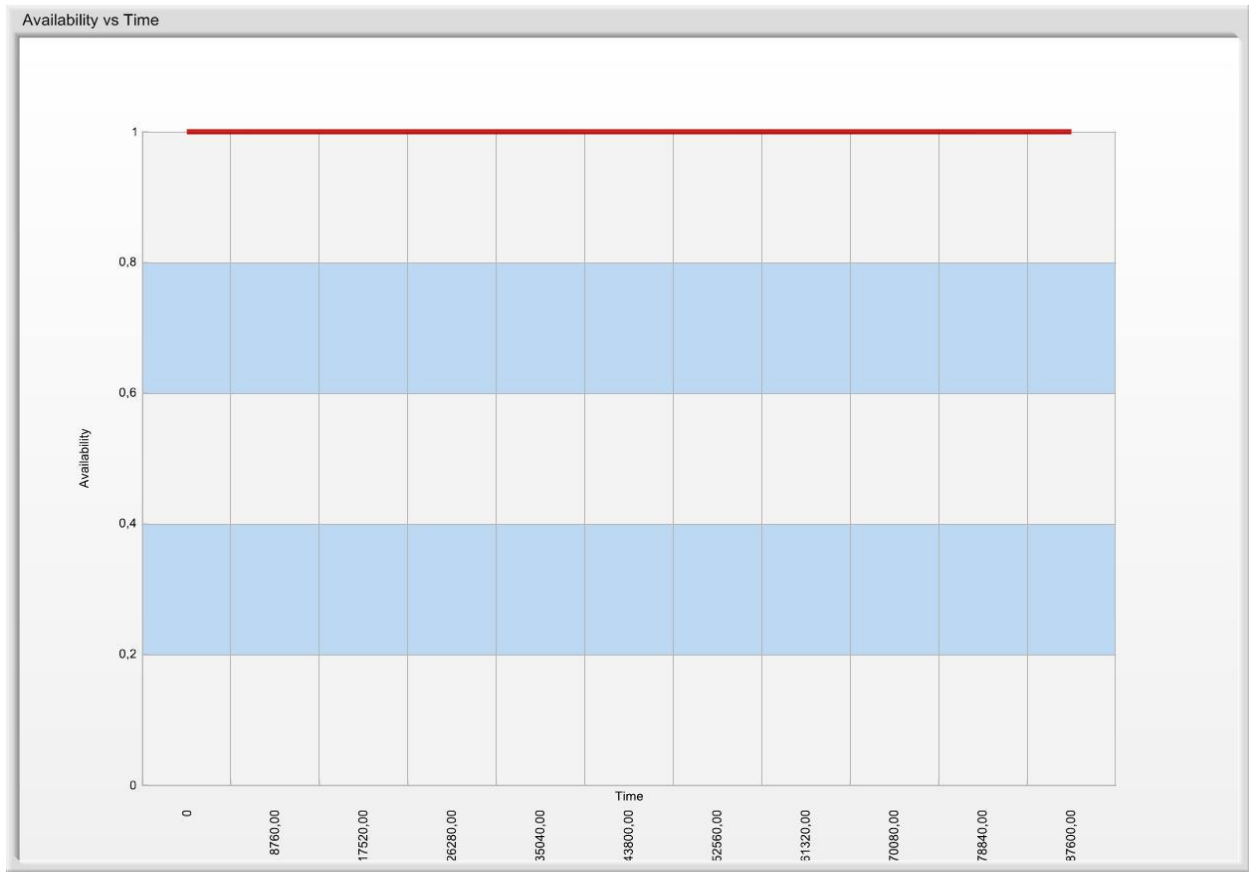
Tabela rezultatov Markovske analize (dosegljivost komponent v različnih časovnih točkah)

| Število let delovanja | Analiza napajanja | Analiza pomnilniških komponent | Analiza V/I komponent |
|-----------------------|-------------------|--------------------------------|-----------------------|
| 0                     | 1,000000          | 1,000000                       | 1,000000              |
| 1                     | 1,000000          | 0,956347                       | 0,996155              |
| 2                     | 1,000000          | 0,916505                       | 0,992325              |
| 3                     | 0,999999          | 0,880142                       | 0,988507              |
| 4                     | 0,999999          | 0,846953                       | 0,984701              |
| 5                     | 0,999998          | 0,816662                       | 0,980908              |
| 6                     | 0,999998          | 0,789015                       | 0,977126              |
| 7                     | 0,999997          | 0,763782                       | 0,973357              |
| 8                     | 0,999996          | 0,740752                       | 0,969600              |
| 9                     | 0,999995          | 0,719733                       | 0,965856              |
| 10                    | 0,999994          | 0,700549                       | 0,962125              |

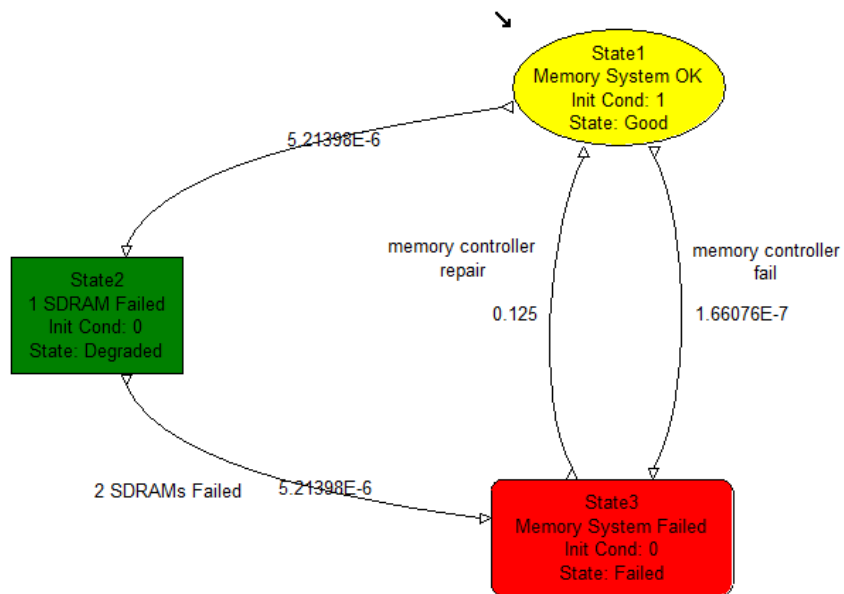
#### Analiza napajanja



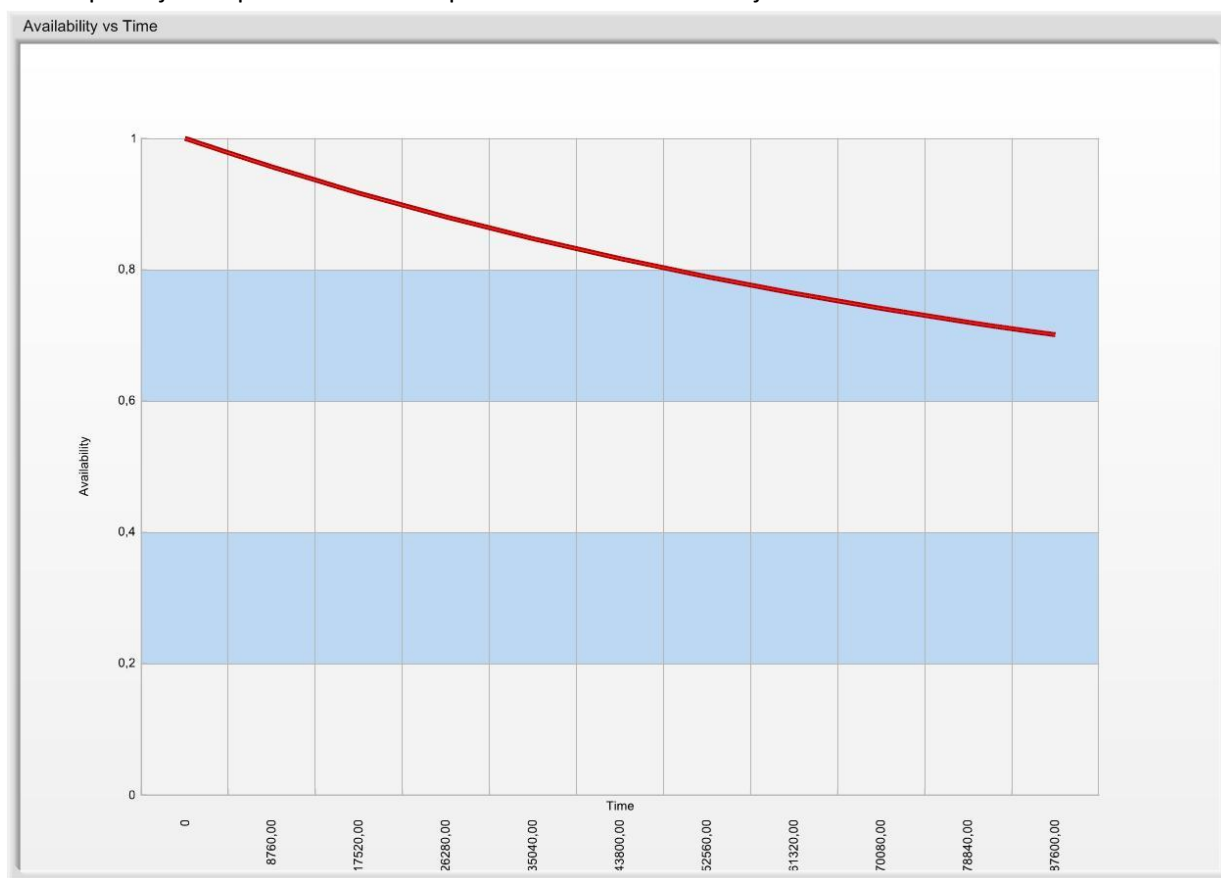
Graf razpoložljivosti napajanja za časovno obdobje 10 let:



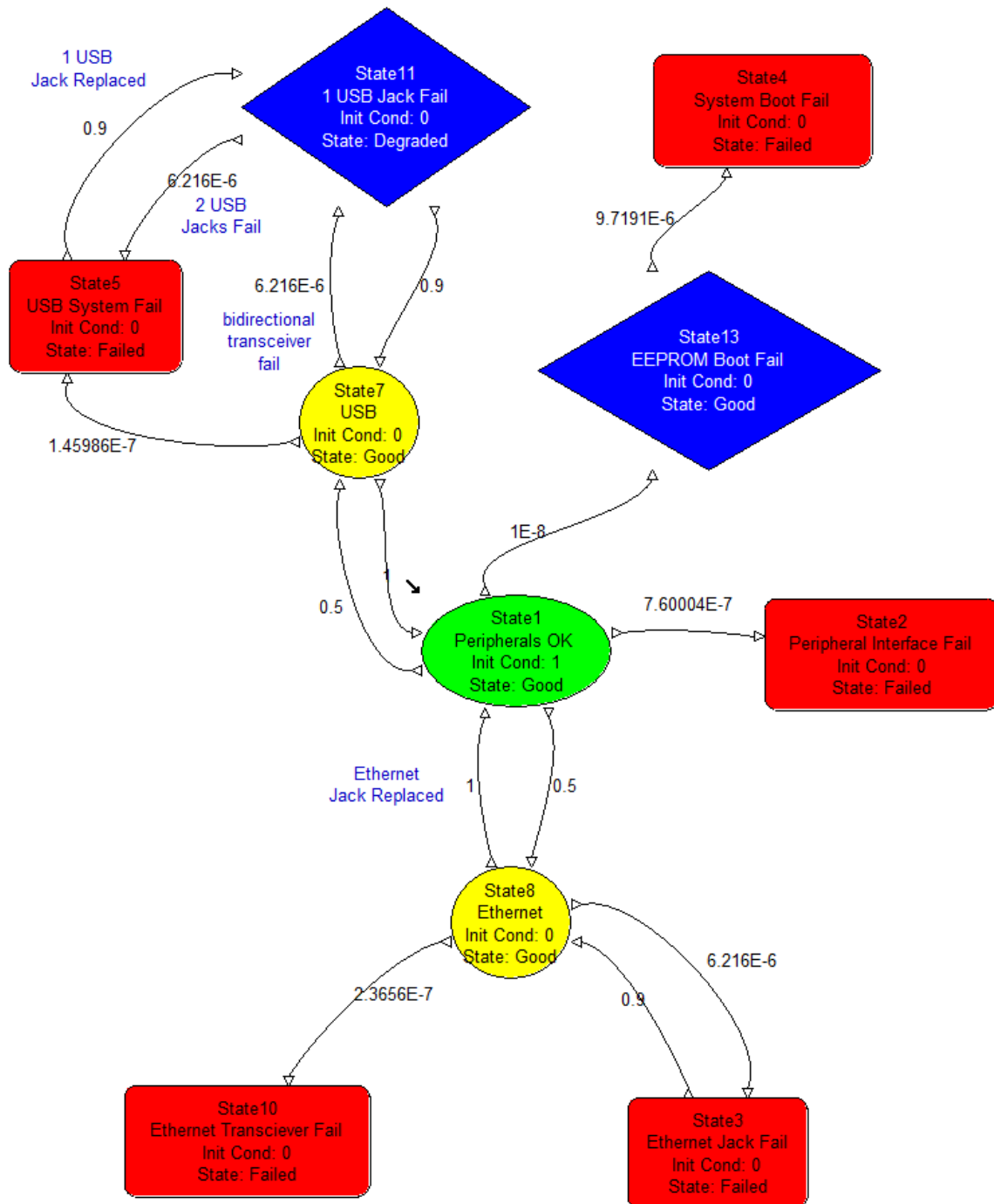
*Analiza pomnilniških komponent*



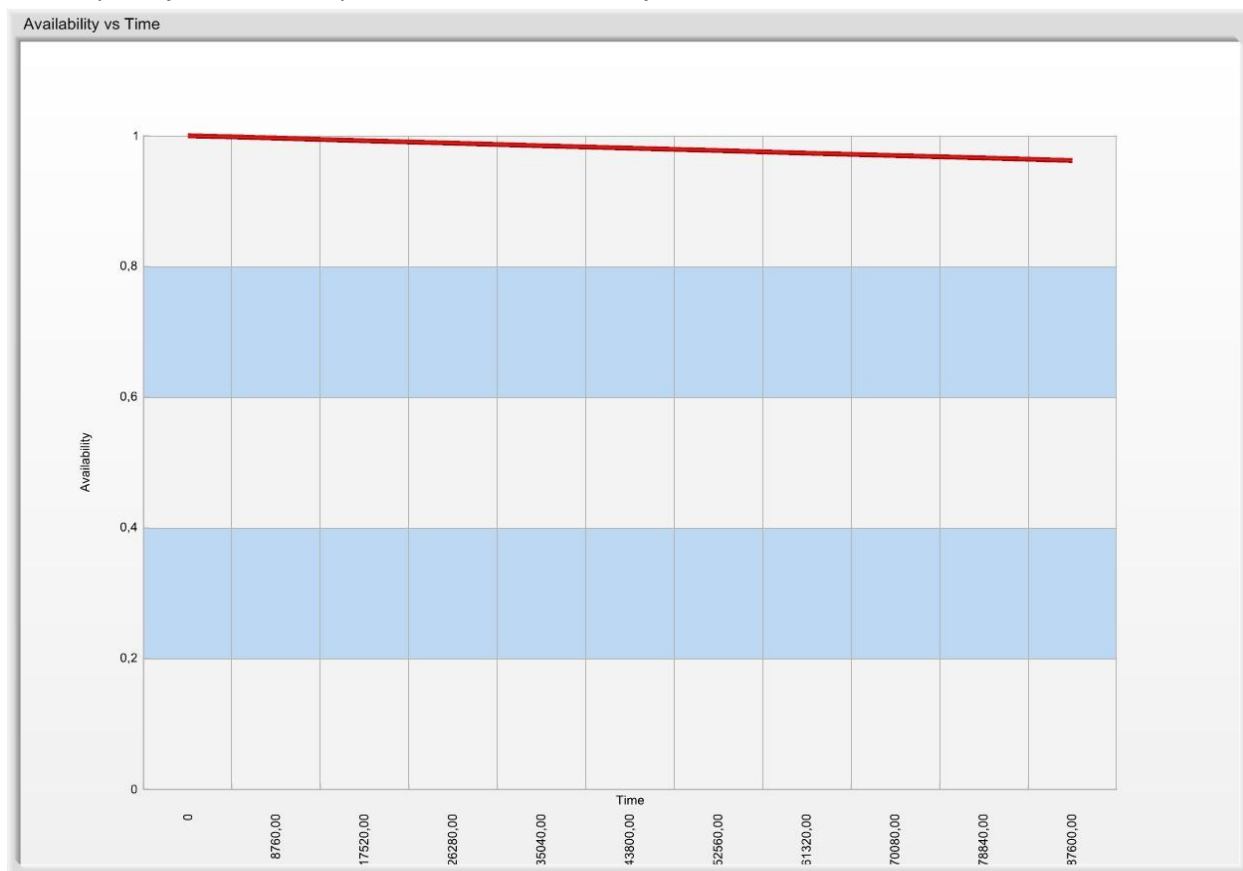
Graf razpoložljivosti pomnilniških komponent za časovno obdobje 10 let:



Analiza vhodno/izhodnih komponent



Graf razpoložljivosti V/I komponent za časovno obdobje 10 let:



## 5.4. Diskusija

Rezultati Markovske analize so precej odvisni od (edinega) parametra  $\lambda$ . V našem primeru je bil še eden ključen faktor kar bi lahko vplival na analizo in to je omejitev brezplačne verzije RELEX Studio, ki omogoča samo 10 stanj v enem projektu in omejuje število povezav med stanji. Rezultati so bolj optimistični, za ta sistem, če nismo odvisni od ETHERNET in USB povezave, kar v našem primeru, ob izpadu enega modula implicira izpad celotnega sistema. Ena velika prednost Markovske analize je dejanje da dobro upošteva odvisnost med komponentami, kjer druge metode(FTA) so zelo slabe. Torej Markovska analiza ni preveč optimistična metoda računalniške zanesljivosti, je bolj realna. Prednosti:

- Prepost pristop modeliranja sistemov
- Tehnike redundance
- Sekvenca dogodkov

Največja slabost pri tej analizi je hitro naraščanje števila stanj z povečevanjem kompleksnosti sistema. To implicira velike grafe, slaba preglednosti in računsko potratne operacije. Na koncu najboljše je če uporabljamo Markovsko analizo v kombinaciji s kakšno drugo metodo, da bi dobili bolj objektivne rezultate. Torej Markovsko analizo uporabljamo za posamezne podsisteme, kjer imamo močno odvisnost med komponentami.



## 6. Literatura

K poglavju 1:

*MIL -HDBK-217 Reliability Prediction of Electronic Equipment , Department of Defense, 1991*

*Dokumentacije komponent: <http://www.alldatasheet.com/>*

*Guidelines to Understanding Reliability Prediction, [www.epsma.org](http://www.epsma.org)*

*Reliability Software & Services from Relx, [www.relx.com](http://www.relx.com)*

K poglavju 2:

*Martin L. Shooman, Reliability of Computer Systems and Networks, 2002, John Wiley & Sons, Inc.*