

# Triple-Modular Redundancy

opracowanie:  
Mikołaj Rybiński

Wydział Matematyki, Informatyki i Mechaniki  
Uniwersytet Warszawski

23 kwietnia 2006

# Plan prezentacji

- 1 Wstęp
  - Błędy w systemach rozproszonych
  - TMR – potrójna, modułarna redundancja
- 2 TMR w CCS
  - Prosty TMR
  - TMR z detekcją błędów
  - SMR – odsiewająca redundancja

# Plan prezentacji

- 1 Wstęp
  - Błędy w systemach rozproszonych
  - TMR – potrójna, modułarna redundancja
- 2 TMR w CCS
  - Prosty TMR
  - TMR z detekcją błędów
  - SMR – odsiewająca redundancja

# Błędy w systemach rozproszonych

## Motywacja

Rozproszony system sam radzący sobie z defektami występującymi podczas jego działania.

## Rodzaje błędów

- Ze względu na źródło:
  - fizyczne np. defekt dysku twardego,
  - projektowe np. błąd programistyczny.
- Ze względu na czas trwania:
  - przejściowe np. wynikające z zakłóceń w polu elektromagnetycznym,
  - trwałe np. uszkodzenie głowicy dysku.

# Radzenie sobie z błędami

## Strategie radzenia sobie z błędami

- Unikanie – np. testowanie i weryfikacja programów.
- Wykrywanie – np. sprawdzanie parametrów funkcji.
- Tolerancja – np. przenoszenie wadliwie wykonywanej funkcjonalności na inne moduły.

## Redundancja komponentów systemu

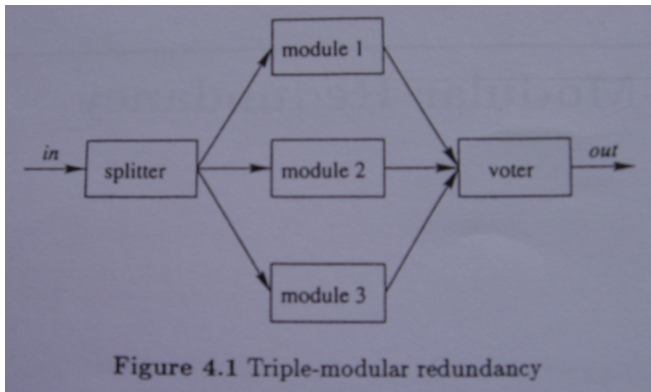
- Technika do wykrywania i tolerancji błędów.
- Niekoniecznie zwiększa niezawodność – logika obsługująca redundancję zmniejsza niezawodność.
- Postulat: jak najprostsza i dokładnie przebadana (wykonana).

# Plan prezentacji

- 1 Wstęp
  - Błędy w systemach rozproszonych
  - TMR – potrójna, modułarna redundancja
- 2 TMR w CCS
  - Prosty TMR
  - TMR z detekcją błędów
  - SMR – odsiewająca redundancja

## Idea

- Wynalazca: John von Neumann.



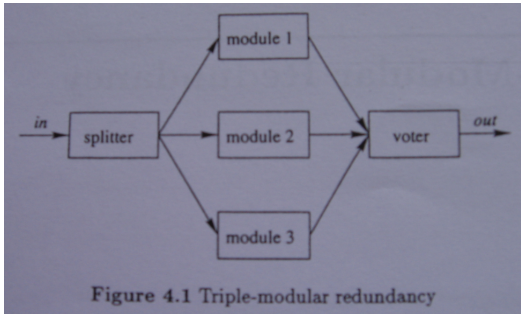
## Możliwości i ograniczenia

- Maskuje zarówno tymczasowe jak i trwałe błędy.
- Warunek poprawności: w każdym punkcie czasu zawodzi co najwyżej jeden komponent.
- Niezależny od modułów wykonujących obliczenia (modularność).
- Nie radzi sobie z błędami projektowymi (zaburzają warunek poprawności).
- Aplikowany przede wszystkim w systemach synchronicznych (globalny zegar), może jednak równie dobrze działać w asynchronicznych.



# Uwagi wstępne

- Podstawowa wersja: 3 komponenty.
- Detekcja błędów: bezpośrednio przez *voter* albo przez osobny moduł (*detector*).
- *Voter* – słabe ogniwo.



# Plan prezentacji

- 1 Wstęp
  - Błędy w systemach rozproszonych
  - TMR – potrójna, modułarna redundancja
- 2 TMR w CCS
  - Prosty TMR
  - TMR z detekcją błędów
  - SMR – odsiewająca redundancja

# Specyfikacja i zarys implementacji

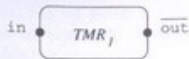


Figure 4.2 The interface of the simple TMR model

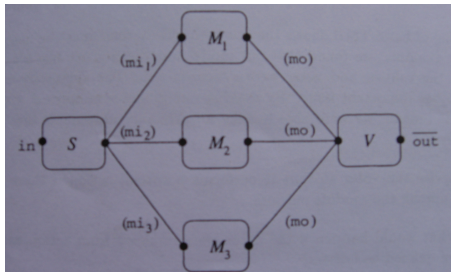


Figure 4.3 The flow diagram of the simple TMR model

# Implementacja i weryfikacja

## Implementacja

Plik: simple\_tmr\_1.vp

## Weryfikacja

```
#bash
./vp simple_tmr_1
#cwb
input "simple_tmr_1.ccs";
eq(TMR1',MP);
dftrace(TMR1',MP);
dfweak(TMR1',MP);
```

# Poprawka i weryfikacja

## Problem

Możliwość buforowania przetwarzanych danych.

## Rozwiązanie

Synchronizacja wyjścia z wejściem. Plik: simple\_tmr\_2.vp

## Weryfikacja

```
#bash
./vp simple_tmr_2
#cwb
input "simple_tmr_2.ccs";
eq(TMR1',MP);
```

# Plan prezentacji

- 1 Wstęp
  - Błędy w systemach rozproszonych
  - TMR – potrójna, modułarna redundancja
- 2 TMR w CCS
  - Prosty TMR
  - TMR z detekcją błędów
  - SMR – odsiewająca redundancja

# Specyfikacja i zarys implementacji

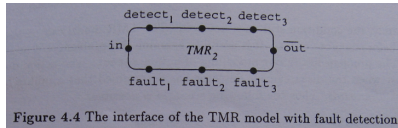


Figure 4.4 The interface of the TMR model with fault detection

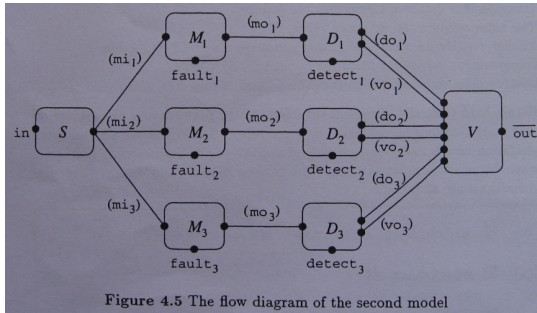


Figure 4.5 The flow diagram of the second model

# Implementacja i weryfikacja

## Implementacja

Plik: detect\_tmr\_1.vp

## Weryfikacja

```
#bash
./vp detect_tmr_1
#cwb
input "detect_tmr_1.ccs";
eq(TMR2',MPU);
eq(TMR2'',Cyclor);
dftrace(TMR2'',Cyclor);
dfweak(TMR2'',Cyclor);
```



# Poprawka i weryfikacja

## Problem

Błędy vs. porażki.

## Rozwiązanie

Rozpoznawanie tylko porażek. Plik: detect\_tmr\_2.vp

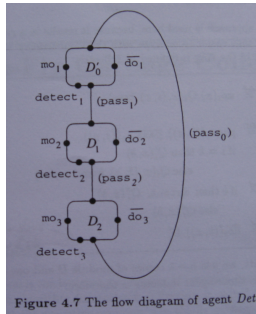
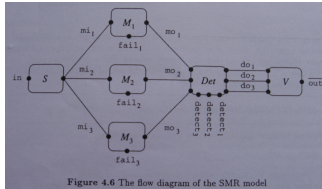
## Weryfikacja

```
#bash
./vp detect_tmr_2
#cwb
input "detect_tmr_2.ccs";
eq(TMR2',MPU);
eq(TMR2'',Cyclor);
```

# Plan prezentacji

- 1 Wstęp
  - Błędy w systemach rozproszonych
  - TMR – potrójna, modułarna redundancja
- 2 TMR w CCS
  - Prosty TMR
  - TMR z detekcją błędów
  - SMR – odsiewająca redundancja

# Zarys implementacji



# Implementacja i weryfikacja

## Implementacja

Plik: smr.vp

## Weryfikacja

```
#bash
./vp smr
#cwb
input "smr";
eq(SMR1,MPU);
eq(SMR2,Cycler);
```