

# UNIVERSITÀ DEGLI STUDI DI ROMA TOR VERGATA

*Dipartimento di Ingegneria Elettronica*

*Sistemi Tolleranti ai Guasti: Generalità ed  
Esempi Applicativi.  
Le Memorie di Massa*



Tor Vergata

SISTEMI TOLLERANTI AI GUASTI: GENERALITÀ ED ESEMPI APPLICATIVI

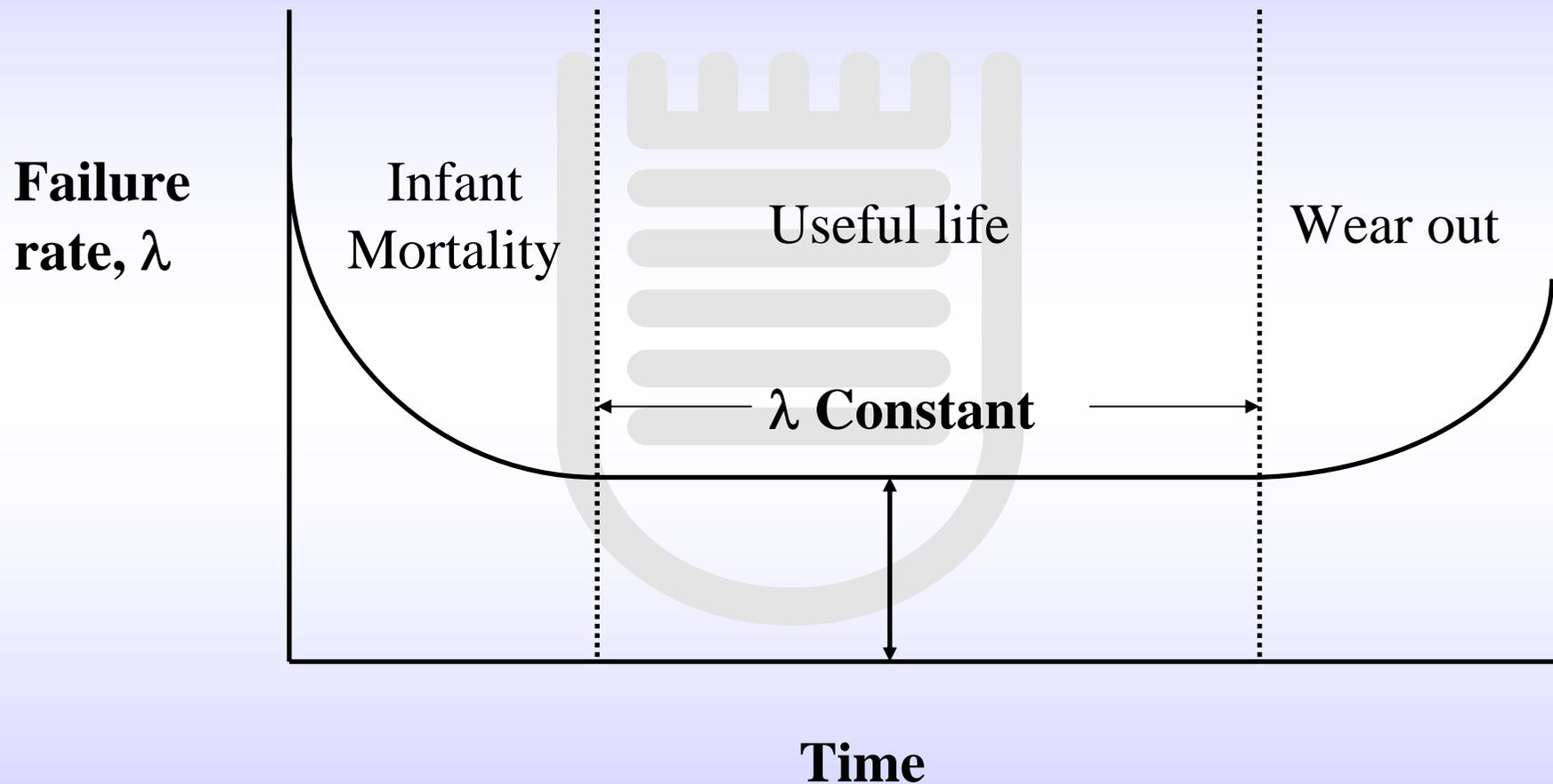


# Svolgimento della Presentazione

- **Principi di base dei sistemi digitali fault-tolerant**
- **Progetto di una Memoria di Massa allo Stato Solido per applicazioni spaziali**
- **Fast – Prototyping della SSMM**
- **Conclusioni**

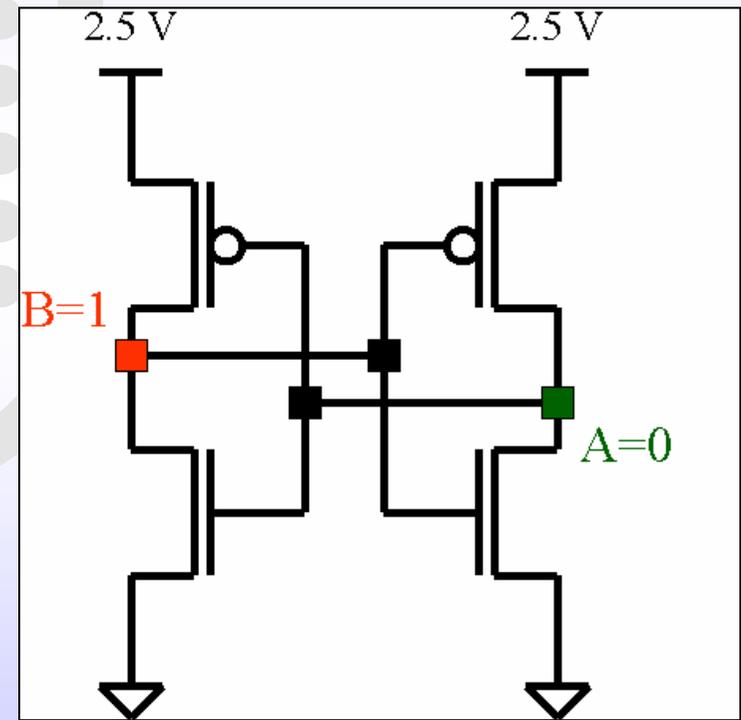
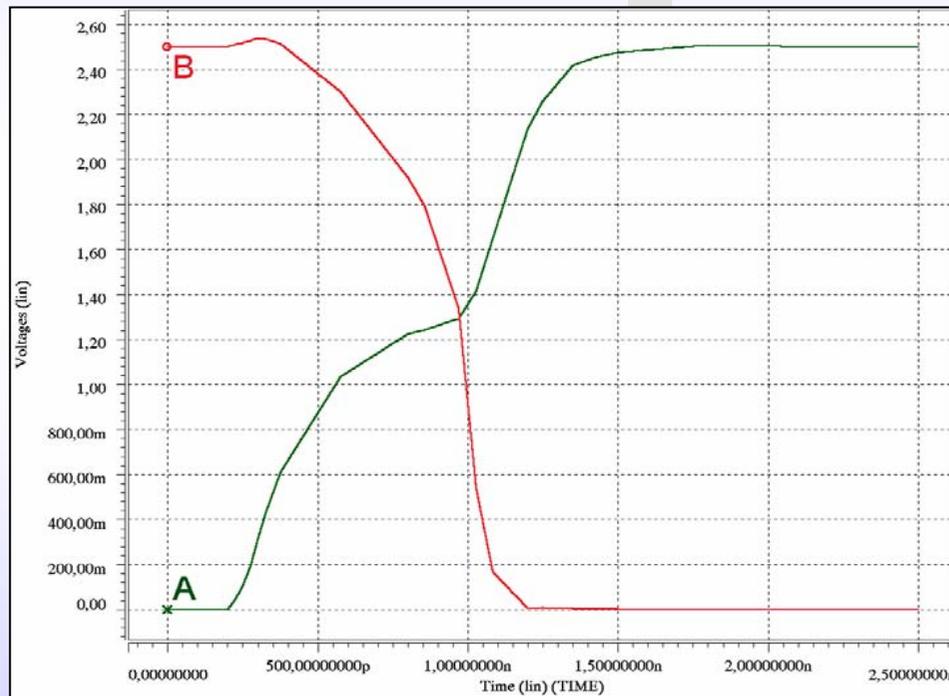


# The Bathtub Curve



# Guasti dovuti alle radiazioni

## Single Event Upset di una cella di memoria SRAM

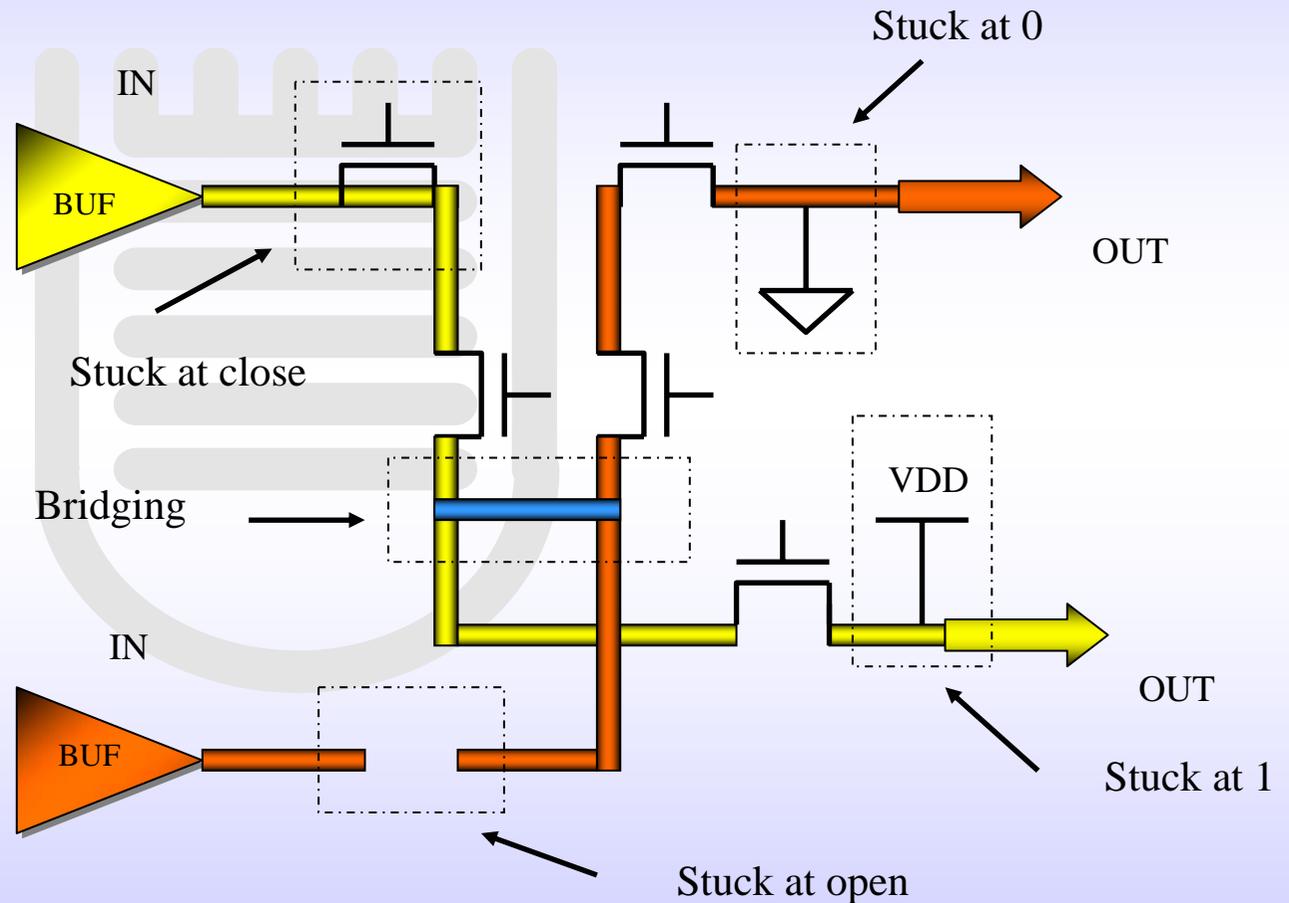


# guasti nelle interconnessioni

● Interconnessioni: costituite da pass-transistors e metallizzazioni.

● Modelli di guasto

- Stuck at 0/1;
- Stuck at open/close;
- Bridging tra linee;



# Aumento dell'Affidabilità

## Aumento della Robustezza del Componente Elettronico

Utilizzano processi tecnologici: SOI, Rad Hard o Space Qualified



## Molto Costosi

Pochi componenti elettronici sono realizzati in questo modo

## Tecniche di Rilevazione e Correzione dei Guasti:

Utilizzo di comp. elettr. allo stato dell'arte (COTS)

Alte prestazioni

Bassi Costi



## Ridondanza

Parte del sistema che non è necessaria per il funzionamento



# Sistemi Self-Checking & Fault-Tolerant

## RIDONDANZA

### STATICA

- ✓ Triple Modular Redundancy
- ✓ Codici a correzione di errore

- ✓ Maggiore Overhead Hardware
- ✓ MTTR = 0  
(Mean Time To Repair)

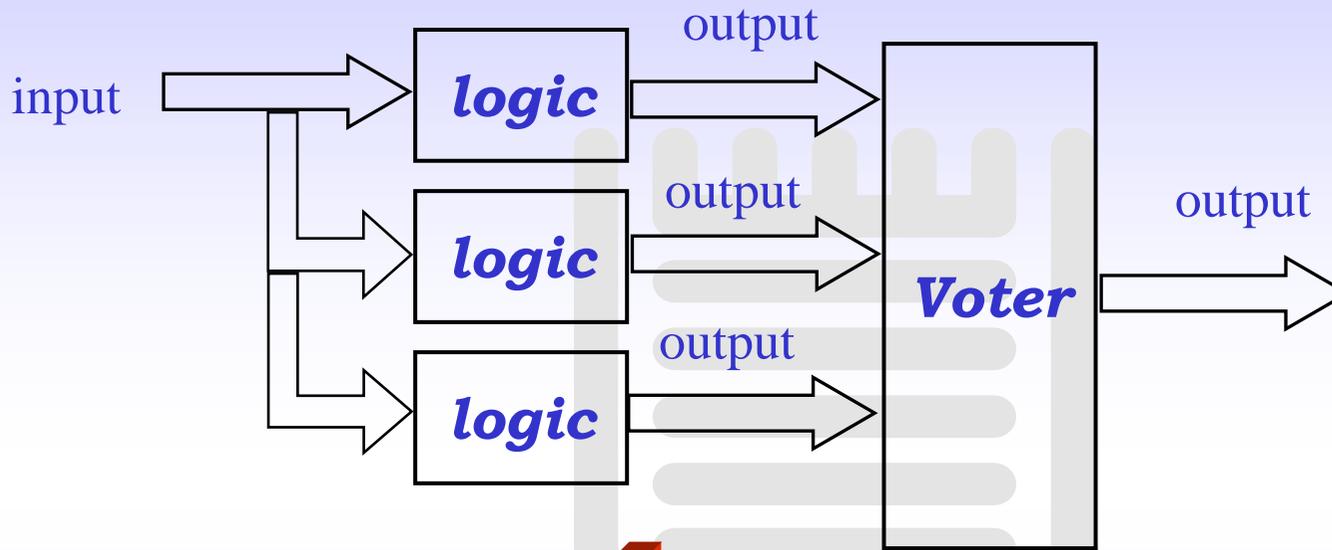
### DINAMICA

- ✓ Fault detection & correction

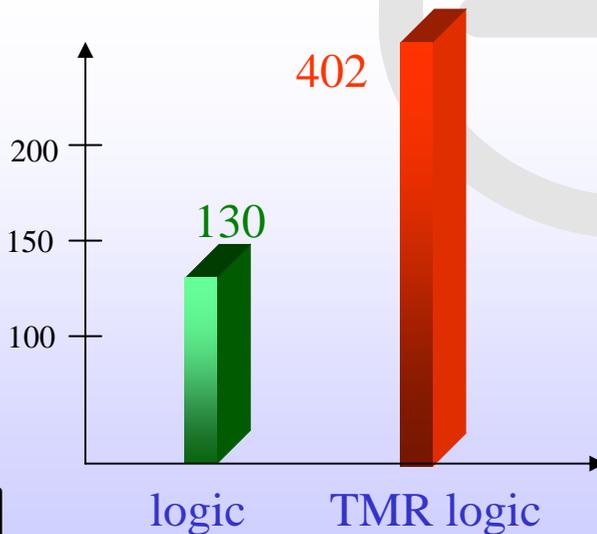
- ✓ Minore Overhead
- ✓ MTTR > 0



# TMR (Triple Module Redundancy)



Area occupata

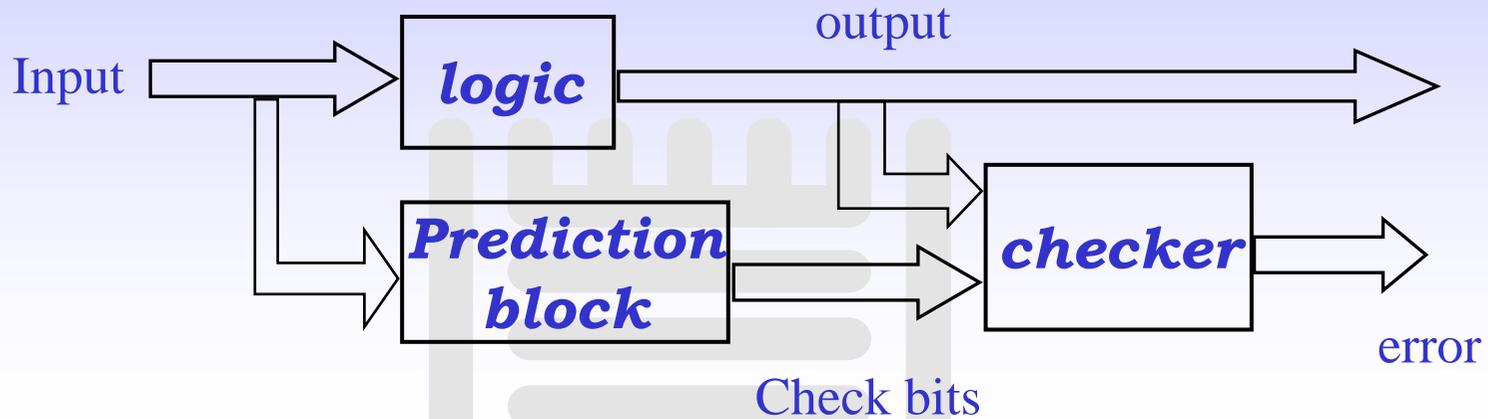


**Alta Ridondanza**

**> 200%**



# Modulo Self-Checking



Area occupata



**Bassa Ridondanza**

**77.19%**

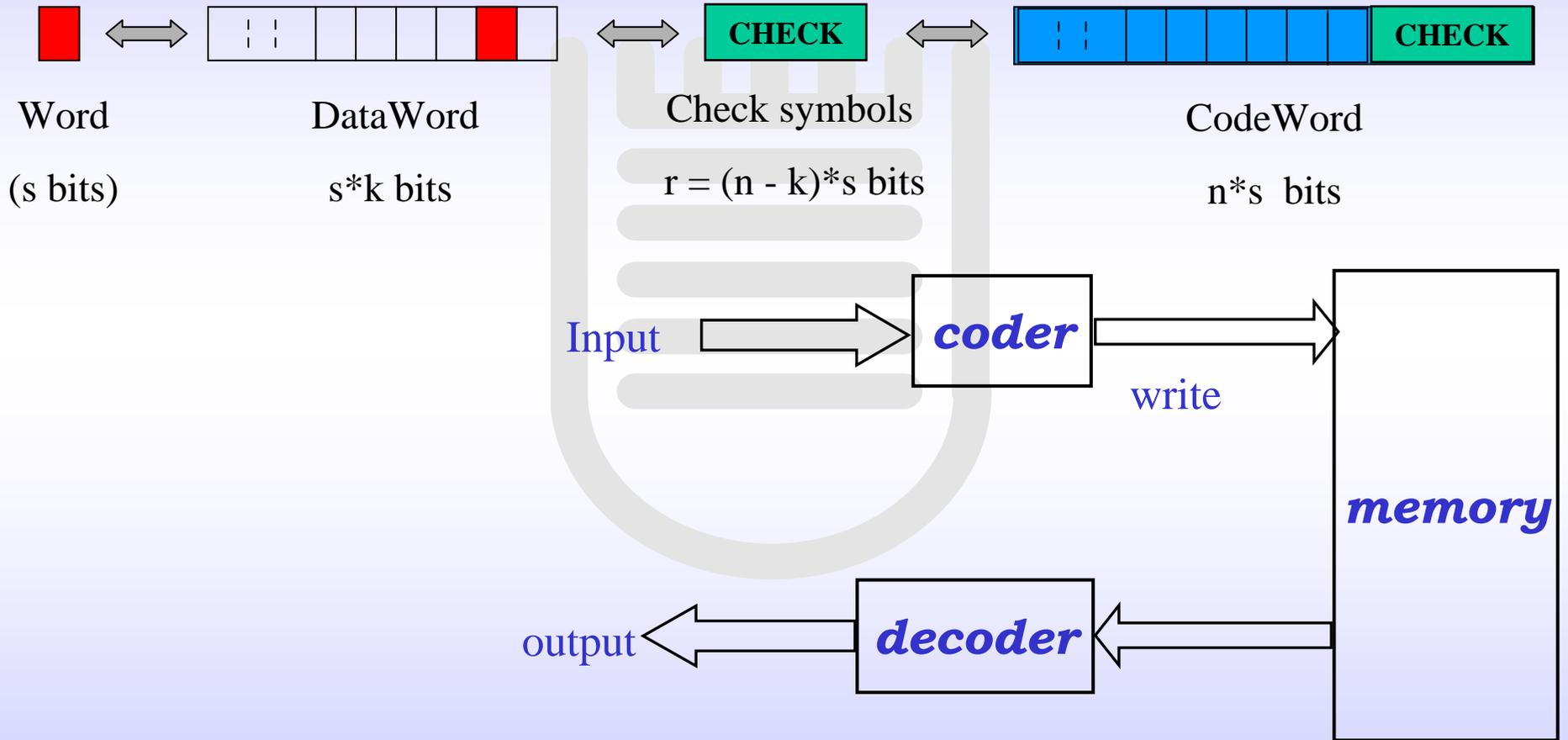


Tor Vergata

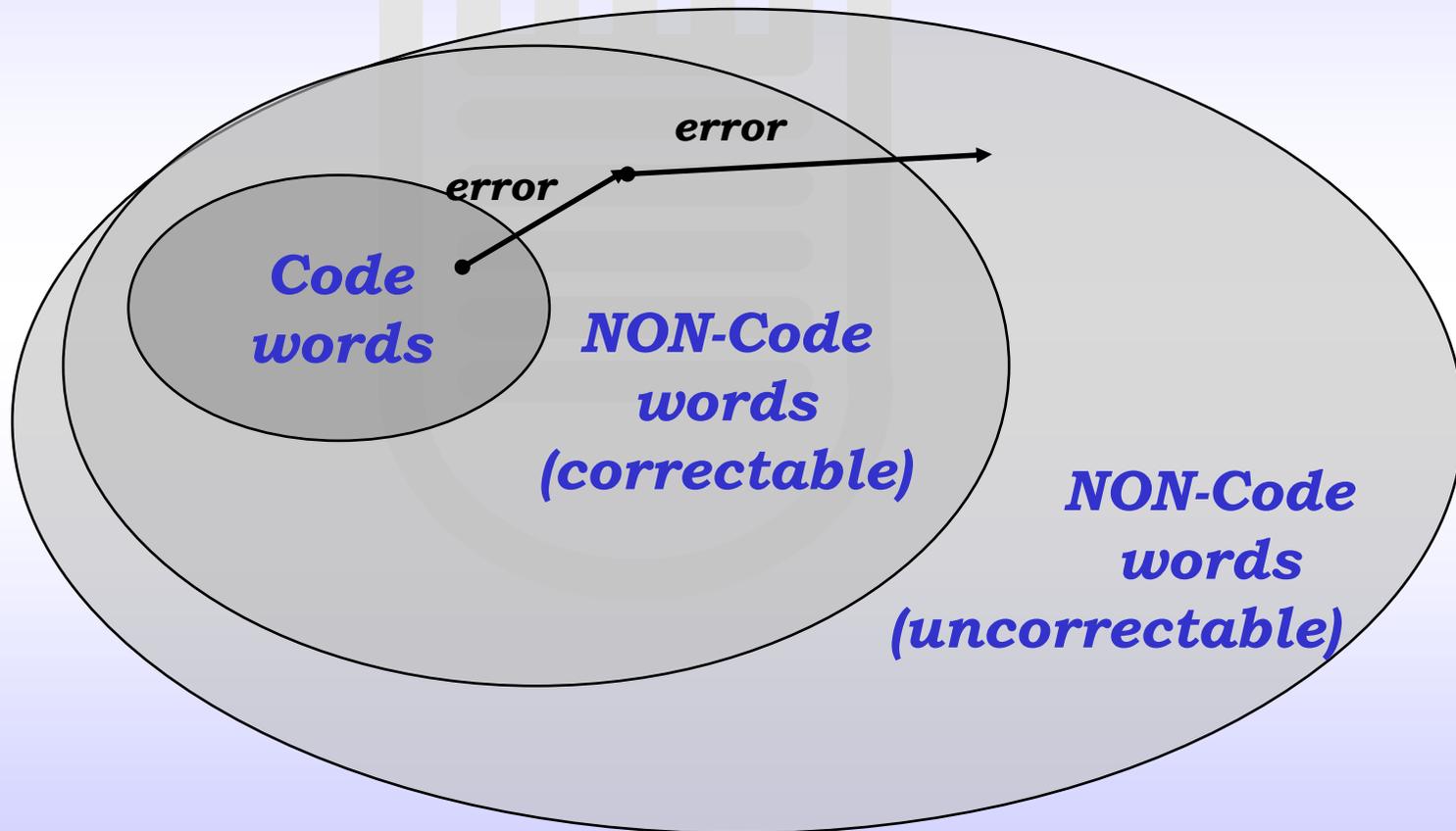
SISTEMI TOLLERANTI AI GUASTI: GENERALITÀ ED ESEMPI APPLICATIVI



# Codici



# Spazio dei Codici



# Codice di hamming

**Hamming code:** error-detecting correcting code  
può:

- (a) Rilevare ogni errore singolo o doppio
- (b) Correggere tutti gli errori singoli

**Matrice di Hamming**

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$

$$\mathbf{d} = [d_1 \ d_2 \ d_3 \ d_4]$$

**Data word**

$$\mathbf{c} = \mathbf{d} * \mathbf{G}$$

$$c_1 = d_2 \oplus d_3 \oplus d_4$$

$$c_2 = d_1 \oplus d_2 \oplus d_3$$

$$c_3 = d_1 \oplus d_3 \oplus d_4$$

$$\mathbf{c} = [d_1 \ d_2 \ d_3 \ d_4 \ c_1 \ c_2 \ c_3]$$

**code word**

# Codici di Reed Solomon

Nei codici di Reed-Solomon  $RS(m,n,k)$  una codeword è composta da  $n$  simboli di  $m$  bits, con una data-word di  $k$  simboli  
Nel caso di  $m=8$  ( 1 byte = 1 simbolo ) si usa la notazione abbreviata  $RS(n,k)$ .

- la correzione avviene sui simboli
- Si possono correggere sia i guasti permanenti (*e.g.* stuck-at nelle celle di memoria) denominati *erasures*, sia i guasti temporanei (*e.g.* SEU) denominati *random errors*
  - Il numero massimo di *random errors* correggibili è  $(n-k)/2$
  - Il numero massimo di *erasures* correggibili è  $(n-k)$
  - In generale si possono correggere  $2*er+re \leq (n-k)/2$  errori



# **OBIETTIVI DEL PROGETTO DELLA MEMORIA DI MASSA ALLO STATO SOLIDO**

- Uso di componenti commerciali in applicazioni spaziali
- Uso di collegamenti seriali basati sul protocollo IEEE 1355.2 (Spacewire)
- Configurazione del sistema in base alla missione
- Riconfigurazione dinamica del sistema di memorizzazione con gestione distribuita del file system
- Fault tolerance: codici EDAC & ridondanza nei moduli di memoria, “graceful degradation” delle funzionalità del sistema



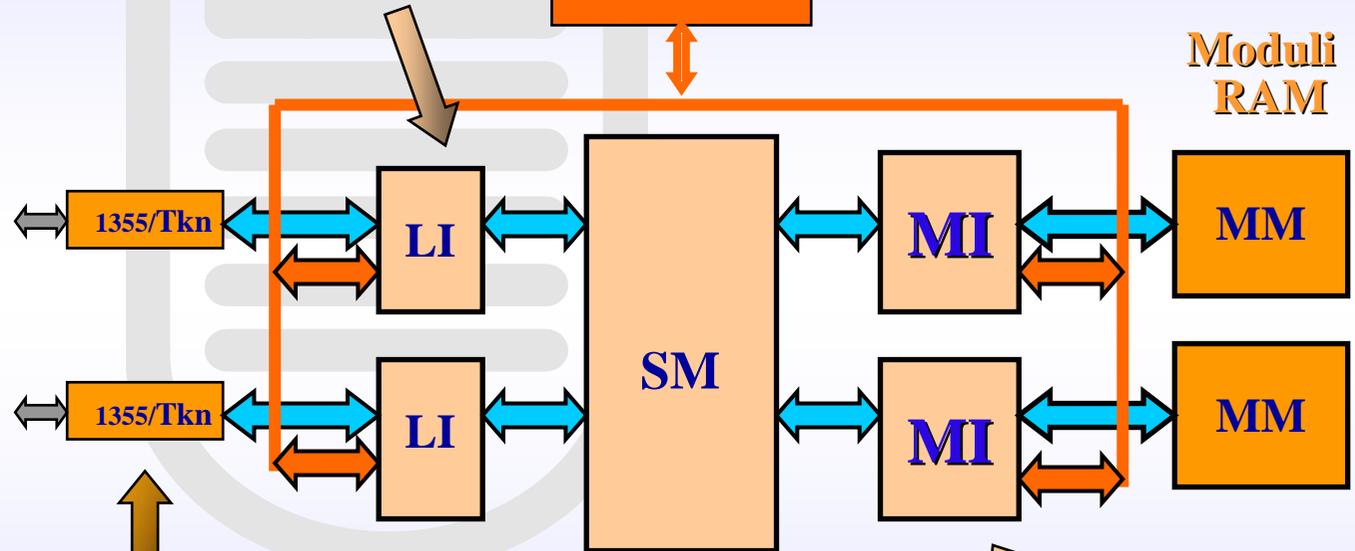
# Struttura della Solid State Mass Memory

## Sistema Di Controllo

Link Interface Gestore  
del routing del pacchetto



Moduli  
RAM



- **Architettura scalabile**

- **Accesso concorrente su moduli di memoria diversi**

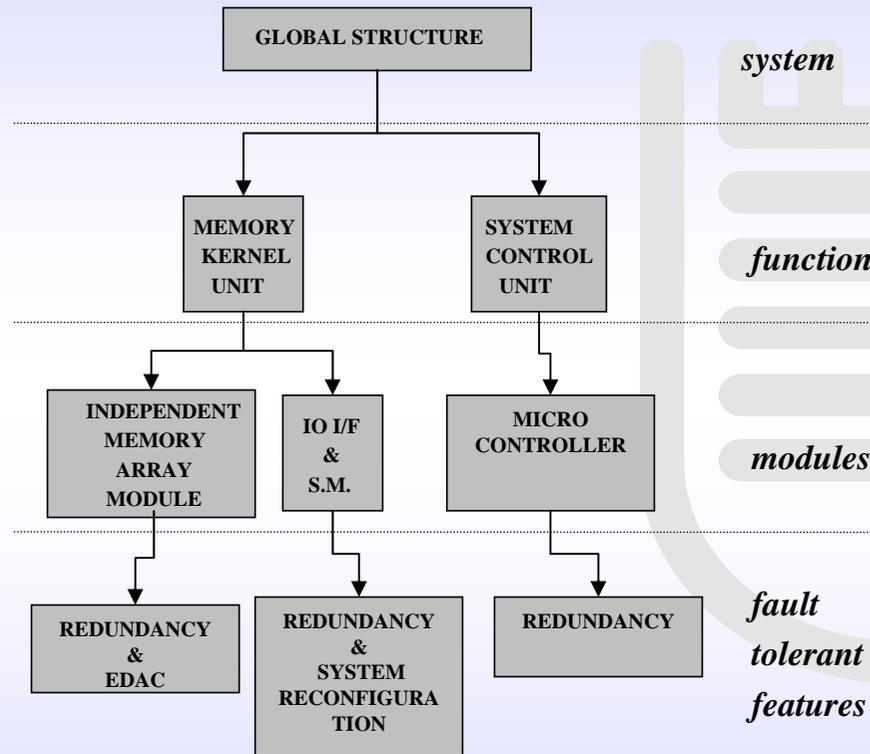
Interfaccia  
1355/Parallelo

Switch Matrix  
Matrice di  
commutazione

Memory Interface  
Interfaccia verso i moduli  
di memoria RAM



# FAULT TOLERANCE DEL SISTEMA

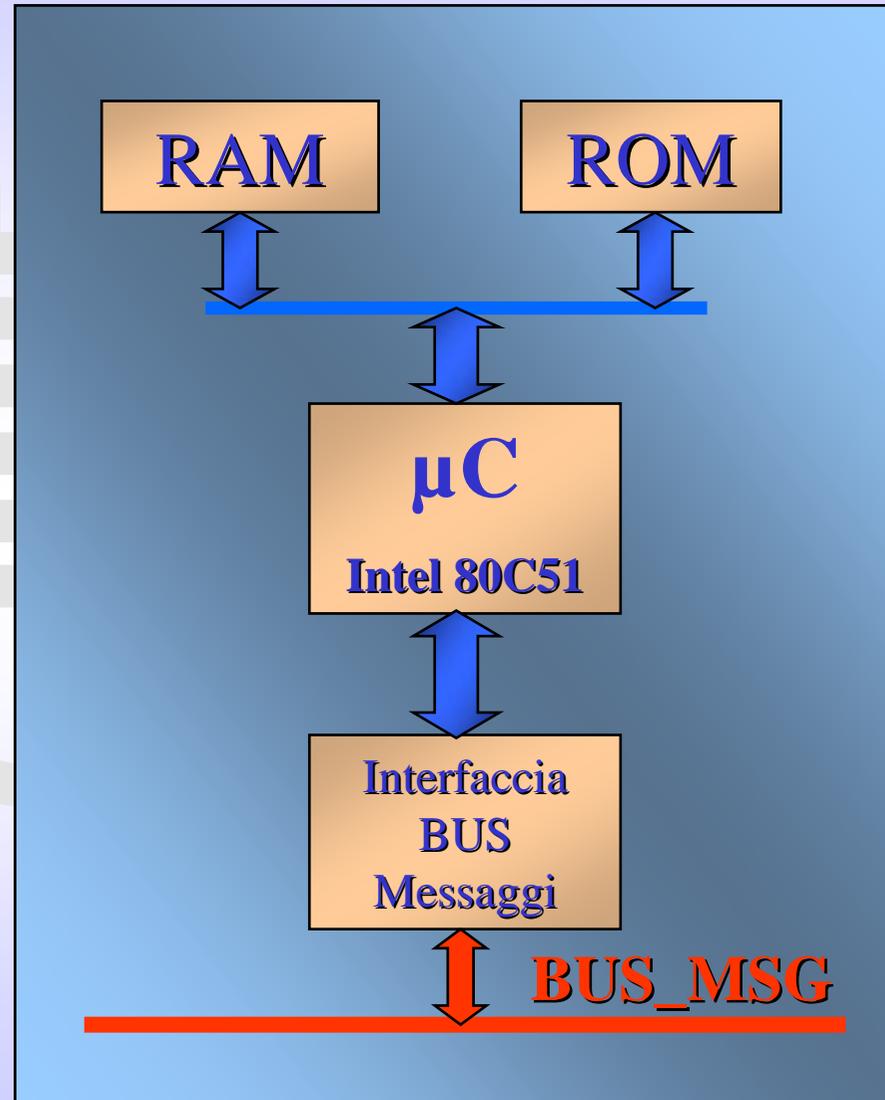


- Ogni modulo è stato progettato con diverse caratteristiche di fault tolerance
- Ridondanza del microcontrollore (SCU)
- Affidabilità nei moduli di memoria con EDAC e ridondanza
- La modularità del sistema di interconnessione permette la riconfigurazione dinamica
- Graceful degradation delle funzionalità di sistema in caso di rotture.



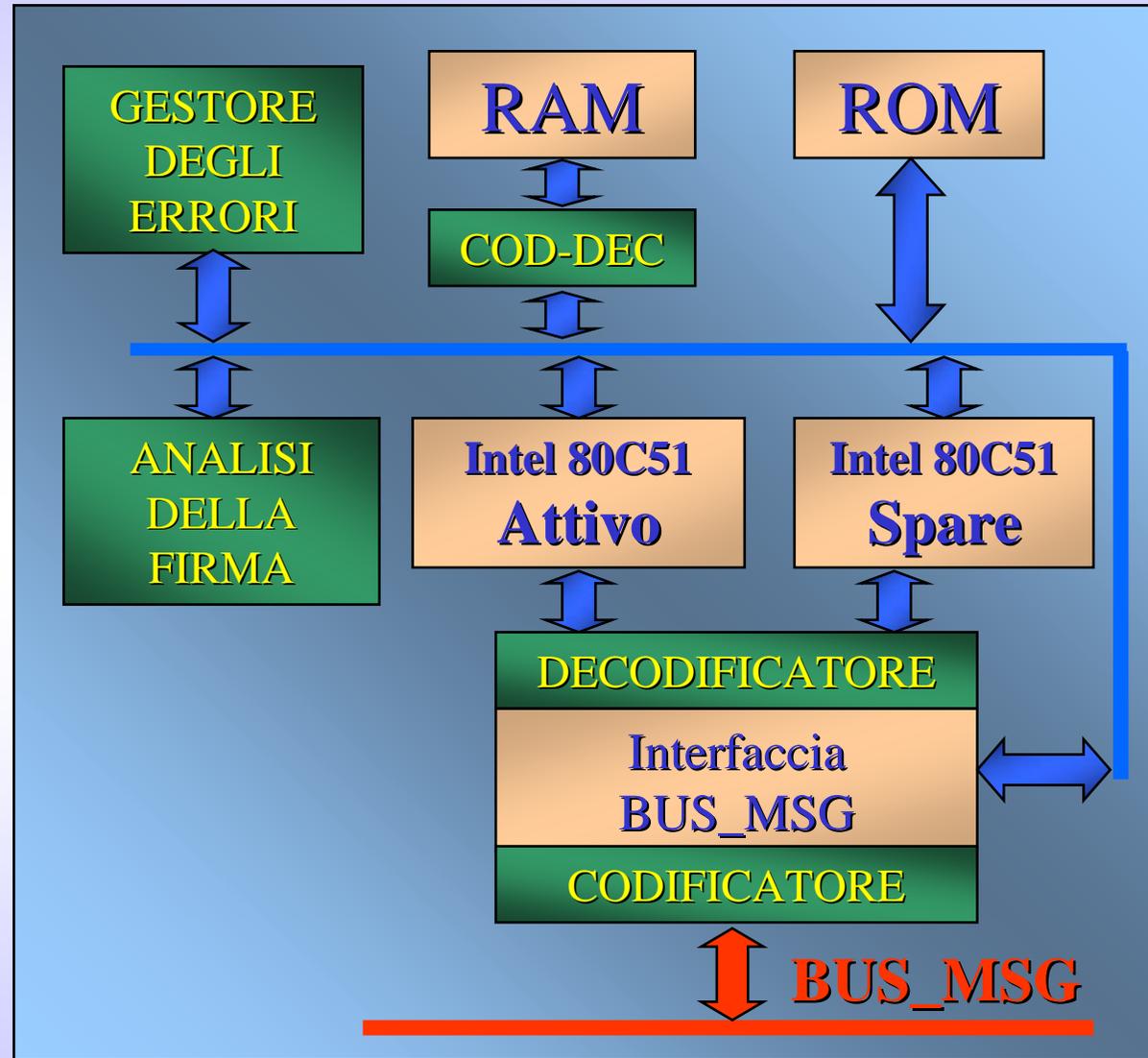
# Sistema di Controllo

- Sistema base senza accorgimenti orientati alla Fault Tolerance.
- Nessuna protezione rispetto a possibili Guasti.



# Sistema di Controllo

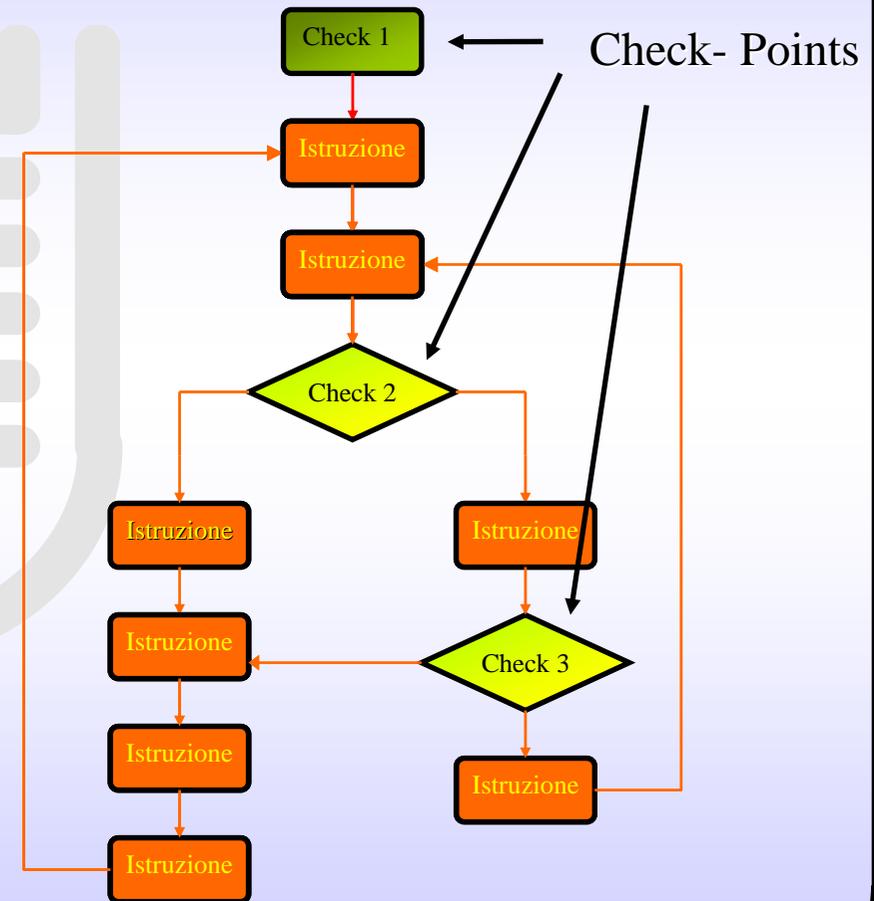
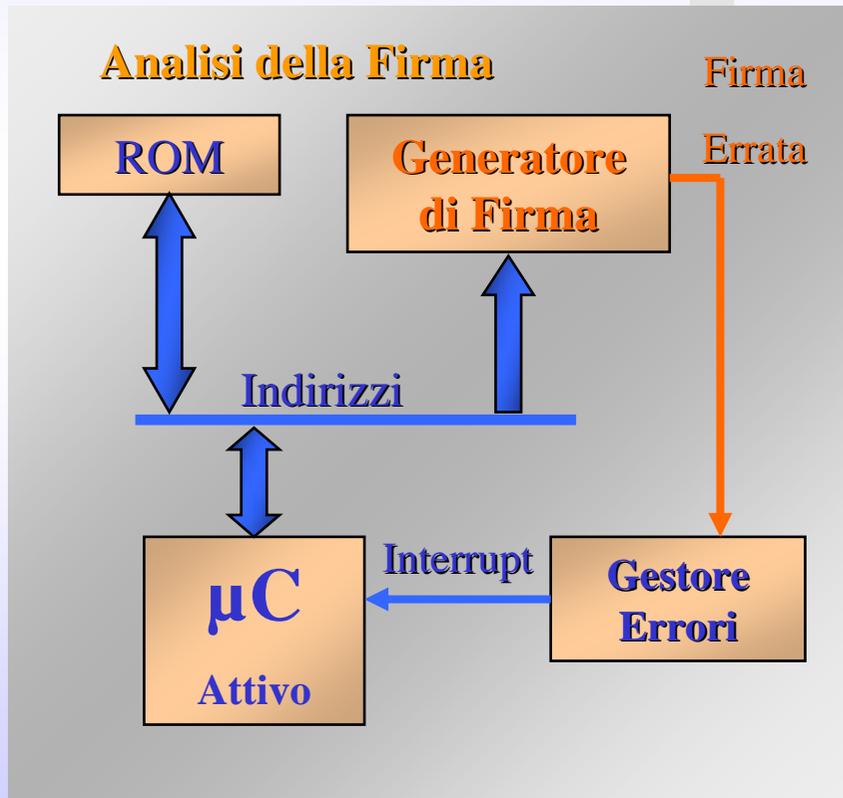
- Duplicazione del  $\mu\text{C}$
- Protezione del  $\mu\text{C}$  tramite Analisi Della Firma.
- Introduzione di un Gestore degli errori.



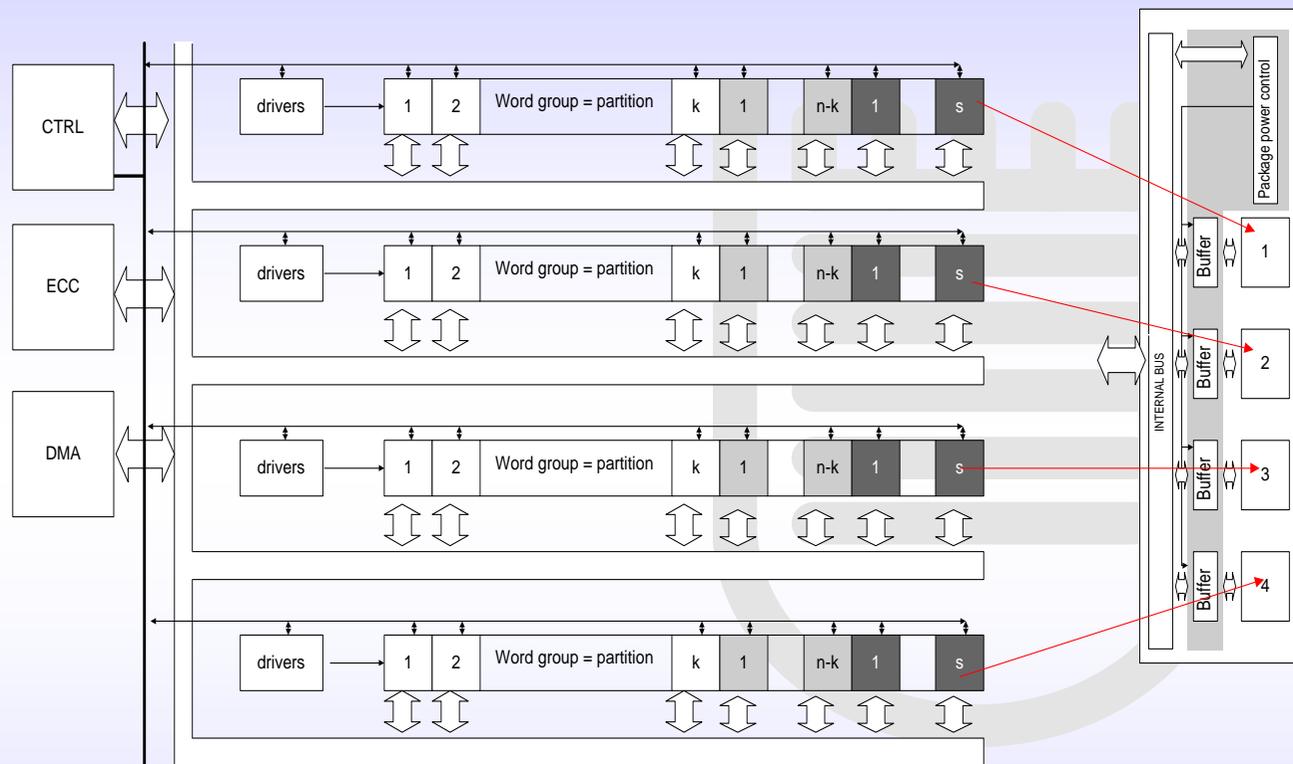
# Analisi della Firma

- Controllo dell'evoluzione del programma
- Indipendenza dal programma
- Controllo Run-Time degli errori

Generico Programma eseguibile  
Analisi delle possibili Sequenze



# Independent Memory Array Modules



4 word group  
4 partition

$k$  = data word length

$n$  = codeword length

$s$  = spare package

- utilizza differenti codici Reed Solomon per ottenere la graceful degradation

- L'ottimizzazione del memory washing permette il Trade Off tra la disponibilità e la codeword length

# Moduli di memoria

INPUT DATA									
Min Value	DECISIONAL VARIABLE	Max Value	Min Value	DECISIONAL VARIABLE	Max Value	Index Selection	PERFORMANCE INDEX DESCRIPTION	Index Weight	
16	SSMM Data Size (GByte)	32	240	Latency Time (hour)	240	<input checked="" type="checkbox"/> S	ECC Inefficiency related to B.E.R.I.R.	1	
9900	SSMM Reliability (1e-4)	10000	0	Spare Package for each Module	10	<input checked="" type="checkbox"/> S	ECC Inefficiency related to R.I.R.	1	
10	SSMM LifeTime (year)	11	0	Spare Memory Module	2	<input checked="" type="checkbox"/> S	Spare Package Inefficiency related to R.I.R.	1	
1000	DRAM Failure Rate (FIT)	1000		TBD		<input checked="" type="checkbox"/> S	Spare Memory Module Inefficiency related to R.I.R.	100	
-15	Bit Error Rate (Log10)	-11		TBD		<input type="checkbox"/> S	Memory Density Inefficiency related to B.E.R.I.R.	1	
784	SEU Rate (upset/Gbit/day)	10737		TBD		<input checked="" type="checkbox"/> S	Memory Density Inefficiency related to R.I.R.	1	
1	Number of Memory Modules	20		TBD		<input checked="" type="checkbox"/> S	Memory Washing Inefficiency related to B.E.R.I.R.	1	
20	Codeword Length	100		TBD		<input type="checkbox"/> S	TBD		
4	Symbol Length (in bit number)	16		TBD		<input type="checkbox"/> S	TBD		
20	Dataword Length	100		TBD		<input type="checkbox"/> S	TBD		
0	Correctable Erasure Number	4		TBD		<input type="checkbox"/> S	TBD		
4	DRAM Number on the Package	4		TBD		<input type="checkbox"/> S	TBD		
64	DRAM's size (Mbit)	64		TBD		<input type="checkbox"/> S	TBD		
0	Memory Washing Efficiency (Log10)	6		TBD		<input type="checkbox"/> S	TBD		
20	EDAC rate (MByte/sec)	60		TBD		<input type="checkbox"/> S	TBD		
70	Code Rate (%)	90		TBD		<input type="checkbox"/> S	TBD		

START OPTIMIZATION AND DISPLAY RESULTS    DISPLAY LAST RESULTS    CLOSE

RESULTS OF OPTIMIZATION PROCESS									
OPTIMIZED VALUE OF DECISIONAL VARIABLE			OPTIMIZED VALUE OF DECISIONAL VARIABLE			OPTIMIZED VALUE OF PERFORMANCE INDEX			
SSMM Data Size (GByte)	23		Latency Time (hour)	240		ECC Inefficiency related to B.E.R.I.R.			0.055
SSMM Reliability (1e-4)	9999		Spare Package for each Module	0		ECC Inefficiency related to R.I.R.			0.00059
SSMM LifeTime (year)	11		Spare Memory Module	0		Spare Package Inefficiency related to R.I.R.			0.00059
DRAM Failure Rate (FIT)	1000		TBD			Spare Memory Module Ineff. related to R.I.R.			0
Bit Error Rate (Log10)	-12		TBD			Memory Density Ineff. related to B.E.R.I.R.			0.21
SEU Rate (upset/Gbit/day)	10737		TBD			Memory Density Ineff. related to R.I.R.			0.0057
Number of Memory Modules	16		TBD			Memory Washing ineff. related to B.E.R.I.R.			0.25
Codeword Length	55		TBD						
Symbol Length (bit number)	8		TBD						
Dataword Length	45		TBD						
Correctable Erasure Number	4		TBD						
DRAM Number on the Package	4		TBD						
DRAM's size (Mbit)	64		TBD						
Memory Washing Efficiency (Log10)	1		TBD						
EDAC rate (MByte/sec)	60		TBD						
Code Rate (%)	82		TBD						

RESULTS READING    CLOSE    COST FUNC. VALUE    0.52

- Capacità dei moduli variabile da 1.125 Gbyte a 4.5 Gbyte
- Codifica Reed Solomon con lunghezza di codeword variabile
- Generazione dei parametri progettuali tramite un tool software originale
- Vincoli progettuali definiti dall'utente finale come input

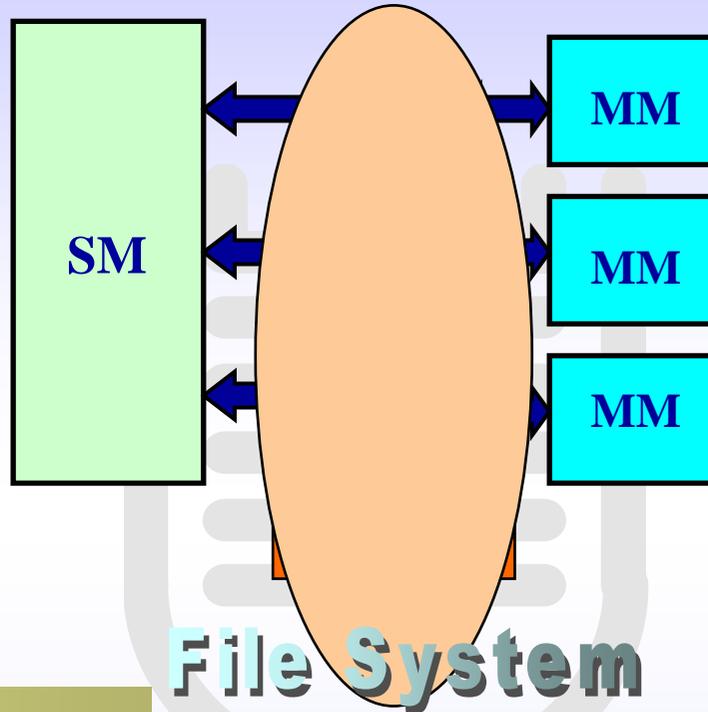
# File System

## Requisiti

- Graceful degradation
- Frammentazione di uno stesso file su più moduli di memoria
- Accesso contemporaneo su più moduli di memoria diversi

## Funzioni agenti sui file

- Scrittura (creazione automatica)
- Lettura
- Cancellazione

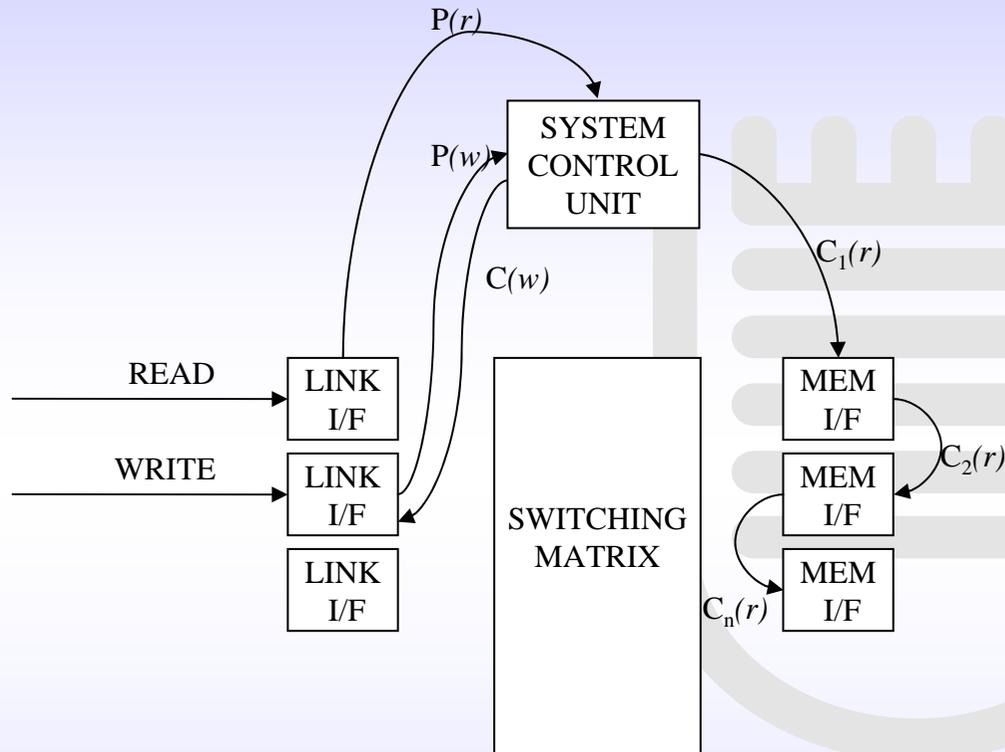


## Soluzione

- Architettura:
  - Distribuita
- Implementazione delle directory:
  - Flat
  - Massimo 256 file
- Metodo di allocazione:
  - Lista concatenata di blocchi
- Amministrazione dei blocchi liberi:
  - Lista collegata



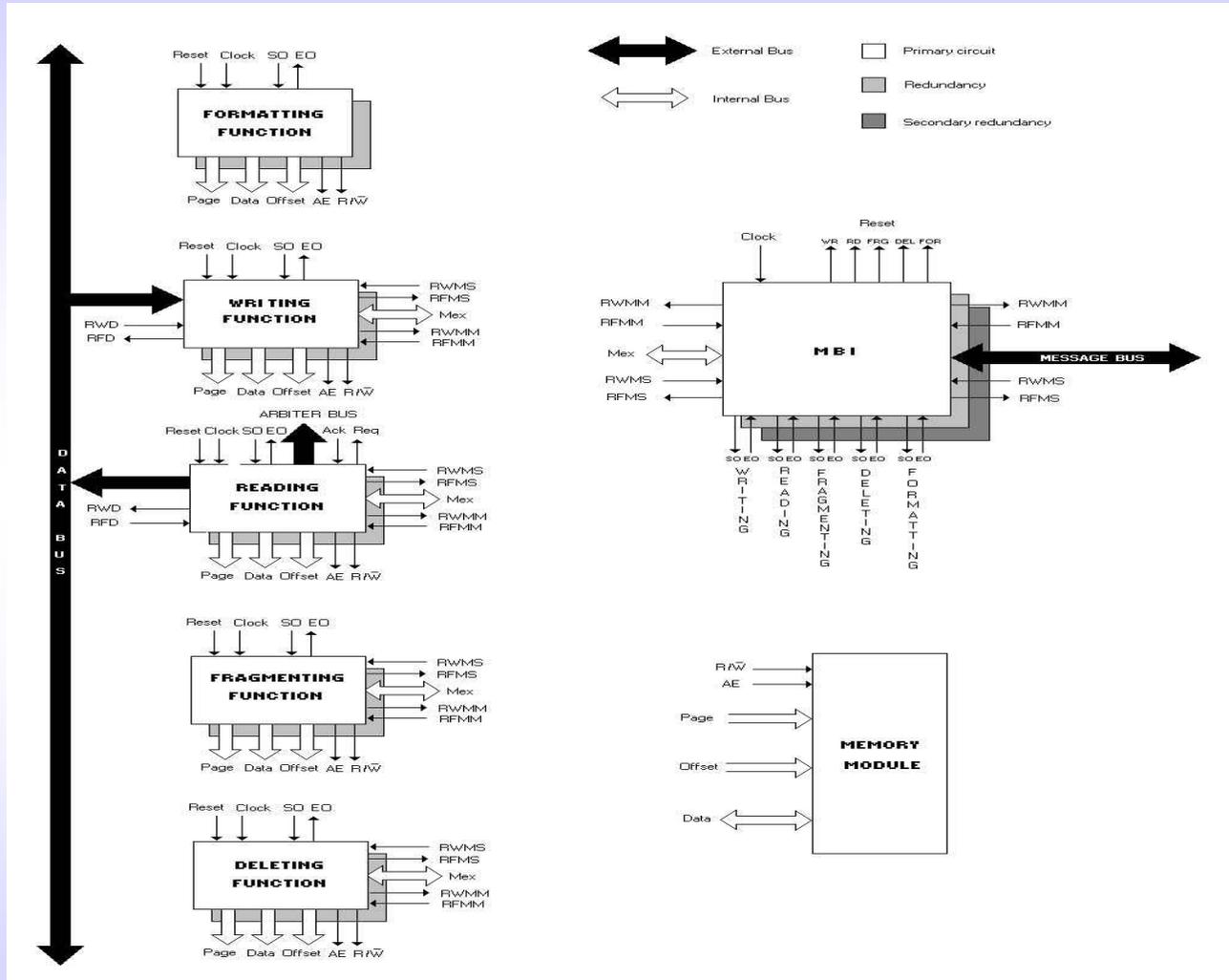
# File System



- Il Microcontrollore (S.C.U.) effettua il “file system management”:
- Operazioni di *READ*
- Operazioni di *WRITE* attraverso i link *spacewire*



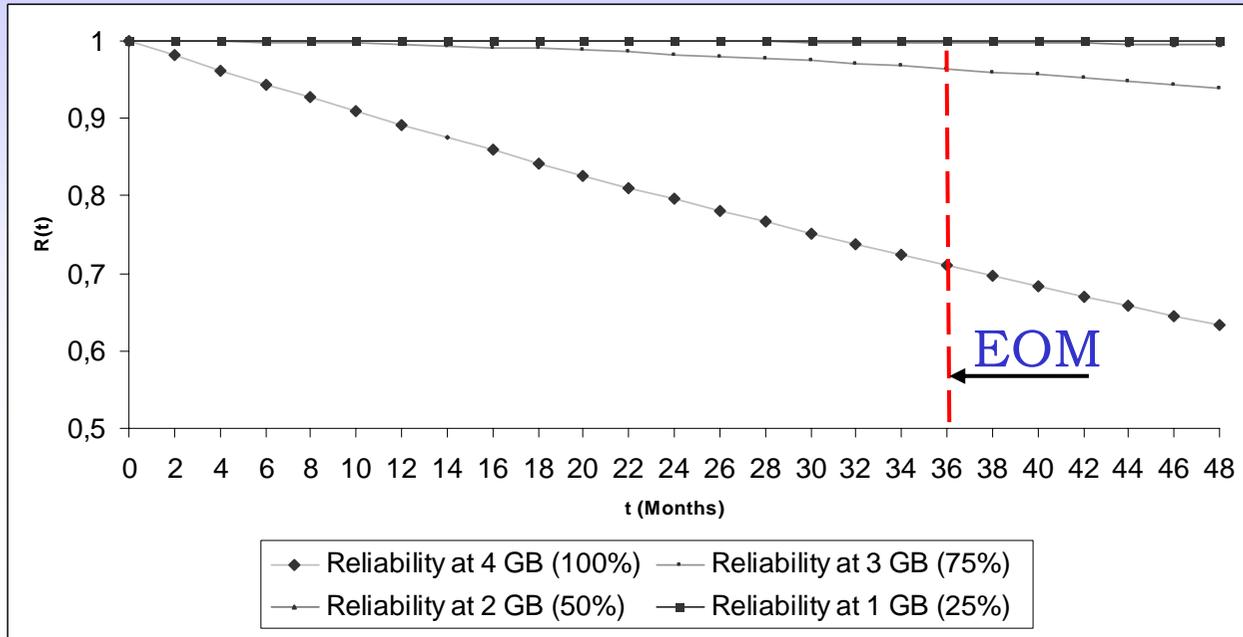
# Memory Interface



- Realizzazione HW delle funzioni di base (lettura, scrittura, cancellazione)

- Graceful degradation delle funzionalità

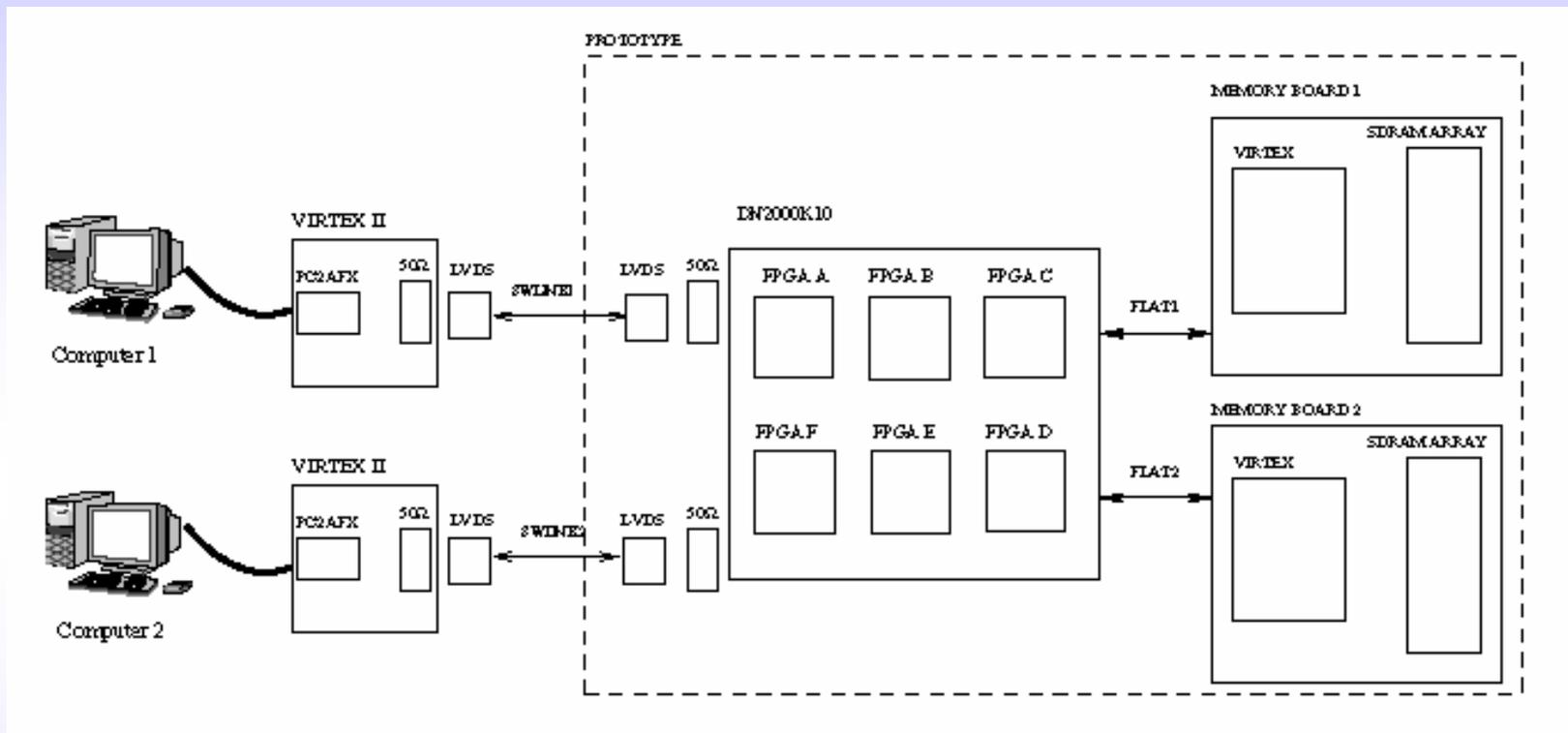
# Gradeful degradation del sistema



$R(t)$ $t=36$ Mesi	4 GB	3 GB	2 GB	1 GB
	0,7097	0,9638	0,9979	0,9999

● Configurazione a Overhead Minimo

# Setup Prototipo



- L'uso di FPGA consente di effettuare campagne di Fault Injection

# FAST PROTOTYPING (1/3)



**Scheda dn2000k10**  
La scheda è un emulatore logico PCI-based per fast-prototyping di ASIC.

equipaggiata con:

- 6 Xilinx Virtex XCV1000
- 1 CPLD per la logica di controllo del funzionamento della scheda
- una memoria flash da 8 MB

**Implementa le I/O interfaces, la SCU e la matrice di switch**

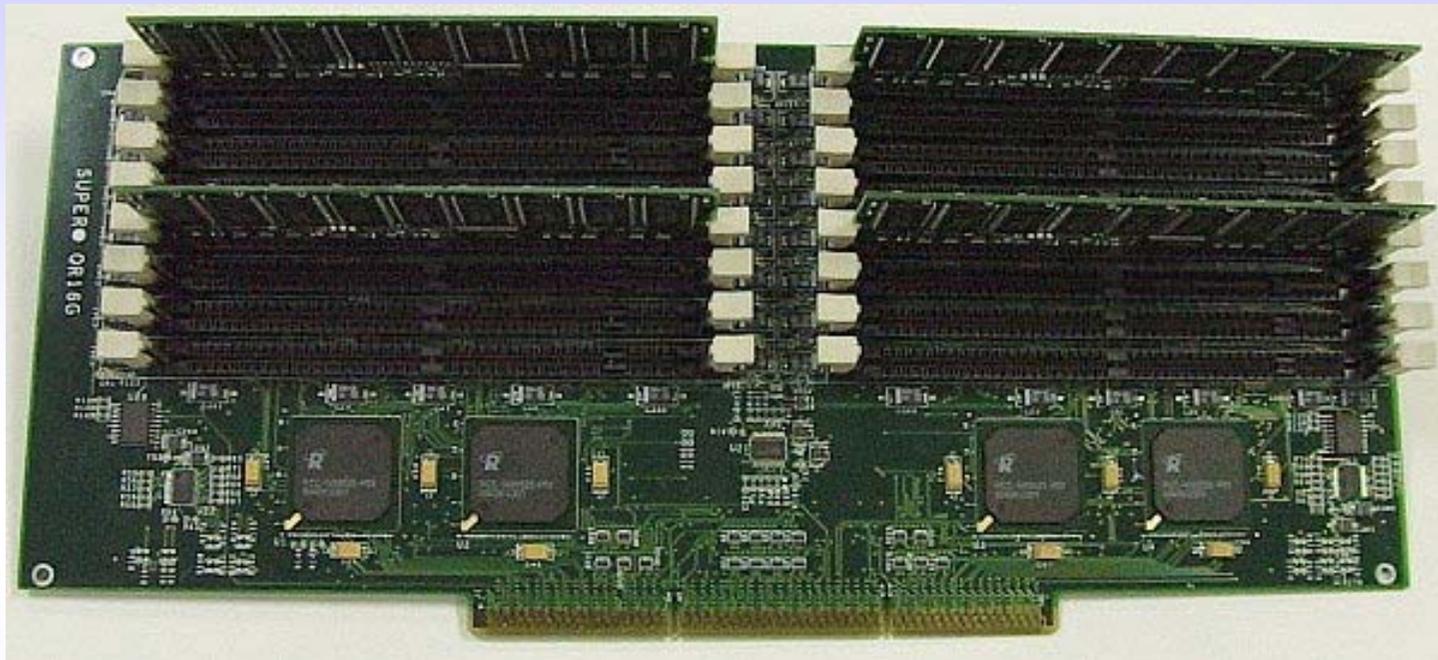


Tor Vergata

SISTEMI TOLLERANTI AI GUASTI: GENERALITÀ ED ESEMPI APPLICATIVI



## FAST PROTOTYPING (2/3)



**La scheda sviluppata permette realizzare un modulo con 4 Gbyte di memoria SDRAM con un CODEC Reed-Solomon**

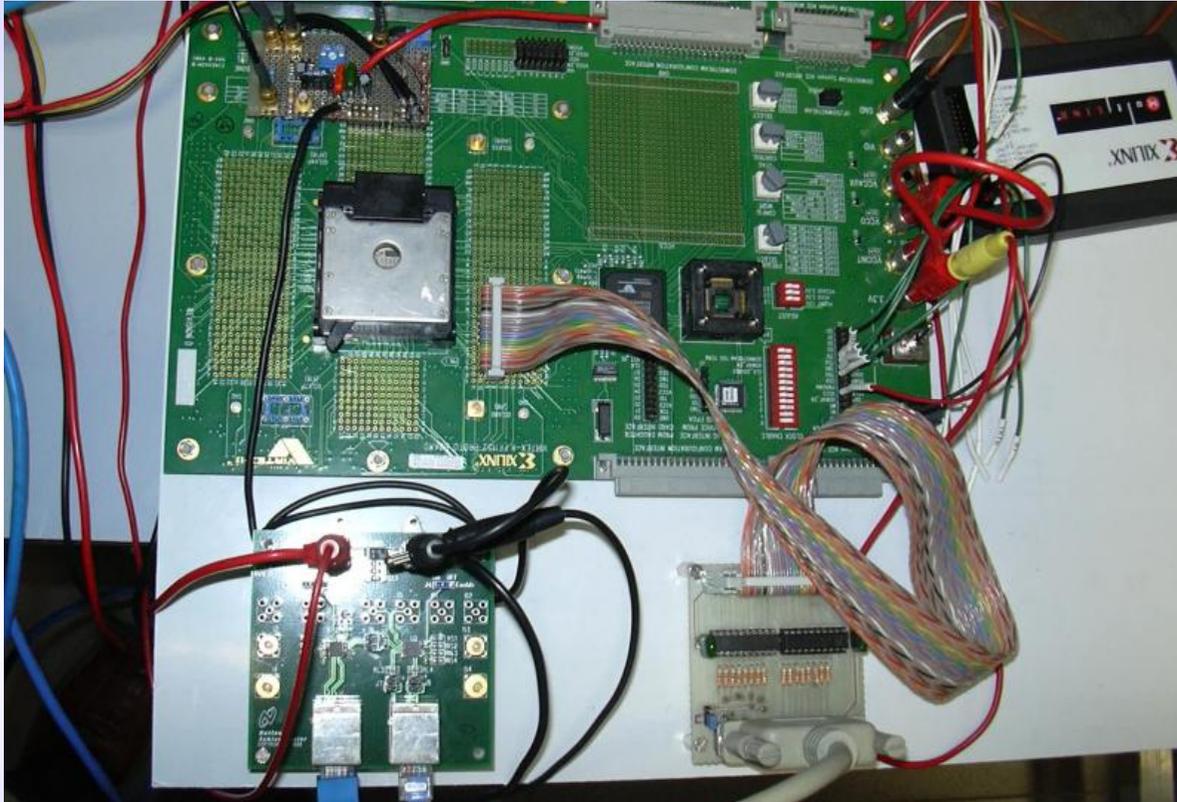


Tor Vergata

SISTEMI TOLLERANTI AI GUASTI: GENERALITÀ ED ESEMPI APPLICATIVI



# FAST PROTOTYPING (3/3)



- Connessione verso PC
- Link SpaceWire
- scheda di conversione LVTTTL - LVDS



# Conclusioni

- L'uso di componenti COTS permette di ottenere prestazioni molto più elevate rispetto a componenti Rad-Hard
- Le metodologie esposte permettono di ottenere alta affidabilità e la graceful degradation del sistema
- Le tecniche di fast-prototyping permettono sia di diminuire i tempi di sviluppo sia di valutare l'affidabilità dei blocchi del sistema

