DYNAMIC FAULT TREE

Emanuele Riccio
M63001339

INTRODUZIONE FTA

- Un **Fault Tree Analysis** è una delle tecniche più importanti utilizzate nell'ambito della reliability e del risk assessment. Lo standard di riferimento è il IEC61025
- La FTA è uno strumento di supporto alle decisioni, in quanto aiuta a trovare le debolezze progettuali e operative in sistemi complessi e a dare priorità ai miglioramenti
- E' un metodo top down e failure based: si inizia da un evento indesiderato(top event) e in maniera deduttiva si ricavano le cause primarie(basic event)
- Si utilizza un diagramma logico denominato **Fault Tree**, per modellare la relazione tra cause e top event. Ogni FT ha un singolo top event per questo è detto single event oriented.
- L'analisi è binaria, dato che ogni evento può accadere o meno, inoltre è deterministico.
- Il FT è un modello che fornisce informazioni qualitative sulle cause del top-event, ma può diventare quantitativo andando a calcolare la probabilità di occorrenza del top event e stimando l'importanza delle cause primarie.

OBIETTIVI FTA

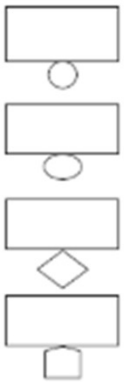
- Identificare tutte le possibili combinazioni di basic event che possono scatenare il top event
- Stimare la probabilità o la frequenza di occorrenza del top event
- Identificare quali parti del sistema vanno migliorate al fine di ridurre la probabilità di occorrenza del top event, tramite una importance metric, si stima il contributo di ogni basic event.
- L'FTA può essere applicata in **sistemi esistenti** sia per identificare debolezze e possibili miglioramenti, sia per identificare le cause di un fallimento del sistema. Per **sistemi in sviluppo** è utile per stimare le probabilità di fallimento ed identificare debolezze di design.

STEP FTA

- 1. Si definisce l'obbiettivo dell'analisi
- 2. Si definisce il top event per cui si andranno ad identificare le cause primarie e calcolare le probabilità. Esso definisce il modo di fallimento che si andrà ad analizzare
- 3. Lo scopo della FTA definisce cosa è incluso nell'analisi, definisce lo stato iniziale del sistema, la versione e le condizioni operative del sistema
- 4. E' definito il livello di dettaglio con cui saranno analizzate le cause di fallimento
- 5. Le ground rules includono le procedure e la nomenclatura da utilizzare per i componenti
- 6. Si costruisce il FT
- 7. Si effettuano analisi quantitative(probabilità top event, dominant cut set) qualitative(minimal cut set) FT.

GATES FTA

PRIMARY EVENT SYMBOLS



BASIC EVENT - A basic initiating fault requiring no further development

CONDITIONING EVENT - Specific conditions or restrictions that apply to any logic gate (used primarily with PRIORITY AND and INHIBIT gates)

UNDEVELOPED EVENT - An event which is not further developed either because it is of insufficient consequence or because information is unavailable

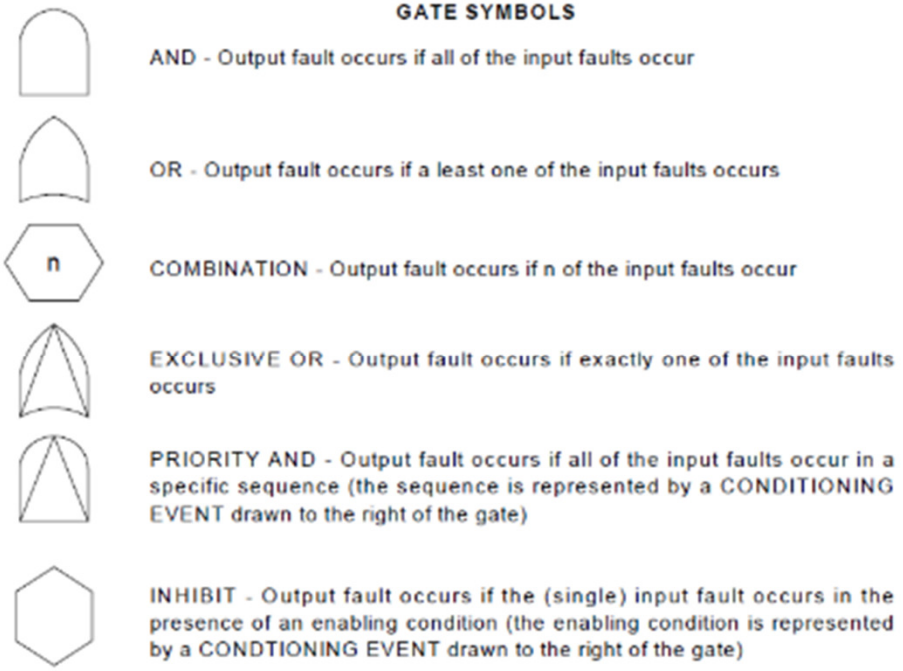
HOUSE EVENT - An event which is normally expected to occur

TRANSFER SYMBOLS

TRANSFER IN - Indicates that the tree is developed further at the occurrence of the corresponding TRANSFER OUT (e.g., on another page)

TRANSFER OUT - Indicates that this portion of the tree must be attached at the corresponding TRANSFER IN

GATE SYMBOLS



AND - Output fault occurs if all of the input faults occur

OR - Output fault occurs if a least one of the input faults occurs

COMBINATION - Output fault occurs if n of the input faults occur

EXCLUSIVE OR - Output fault occurs if exactly one of the input faults occurs

PRIORITY AND - Output fault occurs if all of the input faults occur in a specific sequence (the sequence is represented by a CONDITIONING EVENT drawn to the right of the gate)

INHIBIT - Output fault occurs if the (single) input fault occurs in the presence of an enabling condition (the enabling condition is represented by a CONDTIONING EVENT drawn to the right of the gate)

PRINCIPI FTA

- Immediate, Necessarie, Sufficiente (INS), serve ad identificare le cause INS del top event ad ogni step, l'analisi procede scomponendo ulteriormente le cause identificate
- Primari, Secondari, Comando (PSC), ogni componente ha tre modi di fallimento:
  - Failure Primari: fallimenti indipendenti del componente, che non possono essere dettagliati ulteriormente.
  - Failure Secondi: fallimenti indipendenti del componente, causati da forze esterne al sistema
  - Failure Comando: è un evento atteso o intenzionale che accade in un momento non desiderato a causa di un fallimento
- Stato del componente vs Stato del sistema (SC-SS)
  - Un fault del componente è un fault localizzato in uno specifico componente, si utilizza la logica PSC
  - Un fault del sistema non è localizzato in un componente, ma può riguardare un sottosistema che verrà scomposto ulteriormente con la logica INS

VALUTAZIONE FTA

- Per l'analisi qualitativa viene adoperato l'approccio Minimal Cut Set:
  - Un Cut Set è un insieme di basic event la cui occorrenza simultanea fa accadere il top event. Un CS è minimale è un CS per cui la rimozione anche di un singolo basic event fa perdere lo stato di CS.
  - Il numero di basic event in un MCS è detto ordine, un basso ordine indica che si tratta di una vulnerabilità importante poiché bastano pochi basic event per far accadere il top event.
  - Si ricavano i MCSs grazie all'algoritmo MOCUS. Spesso i CS ottenuti non sono MCS, bisogna ridurli utilizzando l'algebra booleana
- L'analisi quantitativa si basa sull'utilizzo dei MCS ottenuti, i dati dei basic event necessari sono:
  - Failure rate dei componenti
  - CCF
  - Errori umani

COMMON CAUSE FAILURES (CCF)

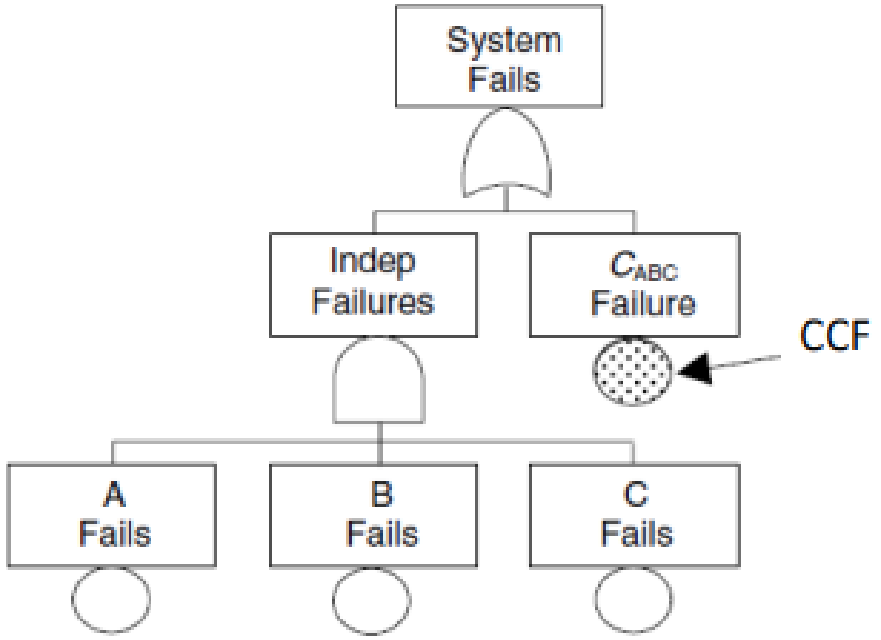
- Un **Common Cause Failure** (**CCF**) è un fallimento di più componenti dovuto a una causa comune durante il funzionamento del sistema
- I progetti dei sistemi sono diventati così complessi che a volte una dipendenza viene inavvertitamente incorporata nel progetto di ridondanza. Una forma di dipendenza è l'**evento CCF che può causare il fallimento di entrambi i sottosistemi ridondanti**
- Poiché l'uso di componenti identici nell'implementazione della ridondanza è comune, i fattori di accoppiamento derivanti dalle somiglianze dei componenti ridondanti sono spesso presenti nei progetti di sistema, con conseguente vulnerabilità agli eventi CCF. Gli eventi CCF di componenti ridondanti identici, quindi, meritano un'attenzione particolare nell'analisi del rischio e della reliability di tali sistemi.
- Se dimentichiamo di includere la possibilità di CCF nelle nostre analisi, **sottostimeremo** in modo significativo la probabilità complessiva di un hazardous event
- Dal punto di vista del FT, ho il top event e una serie di CS. Non considerando i CCF è come se non si considerassero dei CS, quindi si sottostimano i rischi

CCF EXAMPLE

- Consideriamo un sistema con tre componenti identici ridondanti A, B e C che operano in parallelo (un componente che funziona correttamente è sufficiente per il funzionamento del sistema).
- Se ogni componente ha una probabilità di fallimento pari a $\mathrm{p} = 10^{-3}$, la probabilità di fallimento indipendente di tutti i tre componenti è $\mathrm{P}_{\mathrm{indipendente}} = p^3 = 10^{-9}$
- Se si considera la presenza di un CCF che causa il fallimento di tutti e tre i componenti, assumendo la probabilità di occorrenza di un CCF è pari a 10^{-2} ciò equivale a dire che l'1% dei fallimenti è dovuto a CCF.
- La probabilità del fallimento dei tre componenti per CCF è pari a $\mathrm{P}_{CCF} = 10^{-3} * 10^{-2} = 10^{-5}$
- La probabilità di occorrenza del top event è pari a $\mathrm{P}(\mathrm{failure}) = P_{indipendente} + P_{CCF} = 10^{-9} + 10^{-5}$, come si può osservare i CCF incrementano di molto la probabilità di fallimento, si tratta di quasi 4 ordini di grandezza

CCF EXAMPLE

- Per includere il contributo del CCF è sufficiente mettere in OR il CCF con il contributo indipendente dei componenti



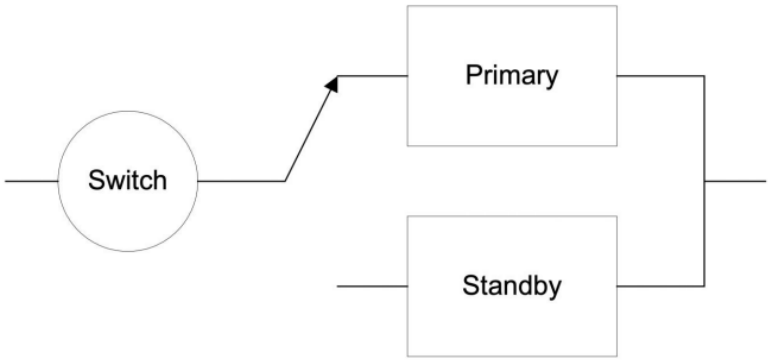
Revised FT model

INTRODUZIONE DFTA

- I Fault Tree statici sono inadeguati quando si vuole modellare sistemi con interazioni dinamiche tra i componenti, dipendenze funzionali e statistiche
- I Dynamic Fault Trees (DFT) estendono i FT classici per permettere di modellare alcune situazioni dove i FT classici falliscono:
  - Sistemi in cui l'ordine in cui si verificano gli eventi influisce sul risultato.
  - Sistemi configurati in modo tale che il verificarsi di un evento provochi l'inaccessibilità o l'inutilizzabilità di altri componenti dipendenti.
  - Sistemi che utilizzano cold spare e sistemi di ridondanza.
- Le tecniche di analisi per i Fault Tree statici non sono applicabili, le metodologie usate per i DFT sono soluzioni algebriche, Modelli di Markov, Petri Nets, Bayesian Network, Monte Carlo simulation

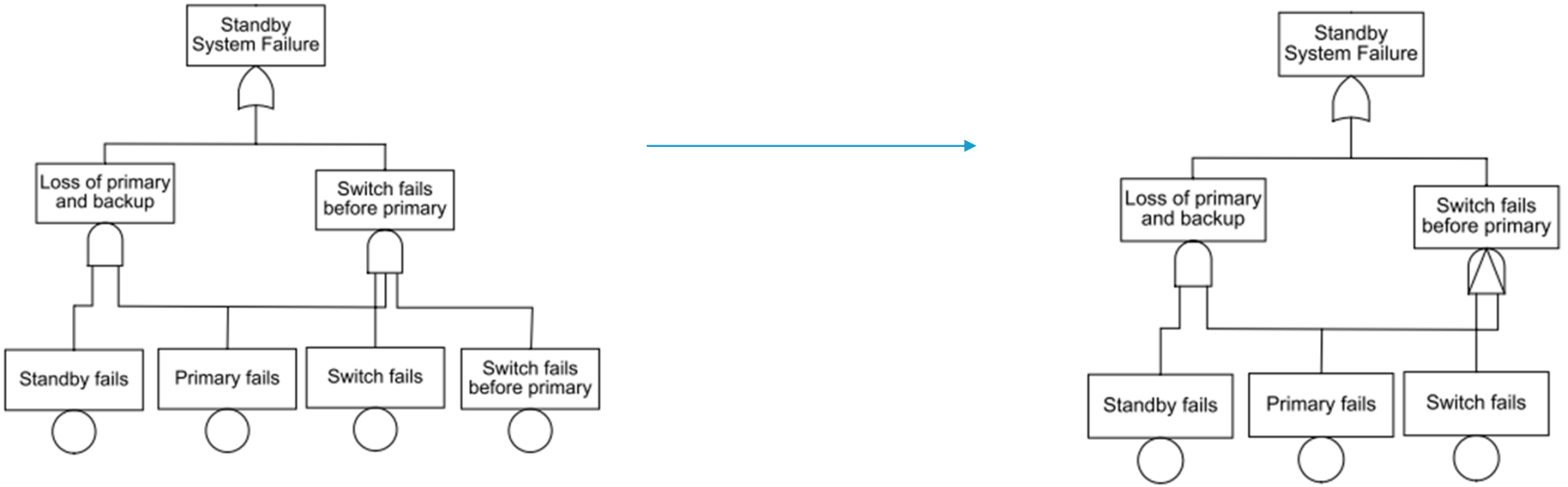
DFT EXAMPLE

- Si consideri un sistema con un componente attivo e uno di riserva collegato a un controllore di commutazione
- L'ordine in cui il primario e l'interruttore falliscono determina se il sistema continua a funzionare
  - Se si rompe prima il primario, effettuiamo lo switch al secondario e poi si rompe lo switch, il sistema funziona
  - Viceversa, se si rompe prima lo switch e poi il primario, il sistema non funziona



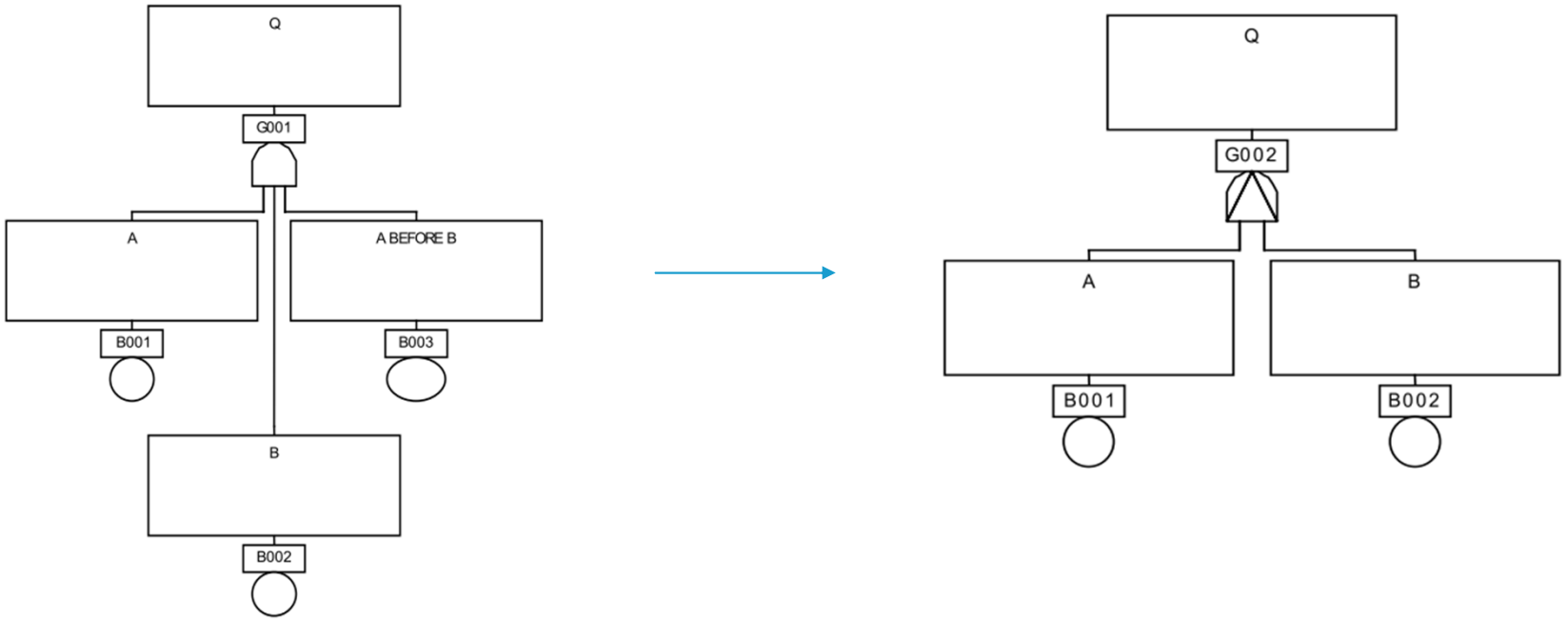
DFT EXAMPLE

- Nel caso di un FT classico, è possibile modellare il sistema con una porta AND esplicitando la condizione temporale. Grazie all'approccio DFT è sufficiente adoperare una porta **Priority-AND**
- I DFT introducono diverse tipologie di dynamic gate estendendo il modello di FT statici



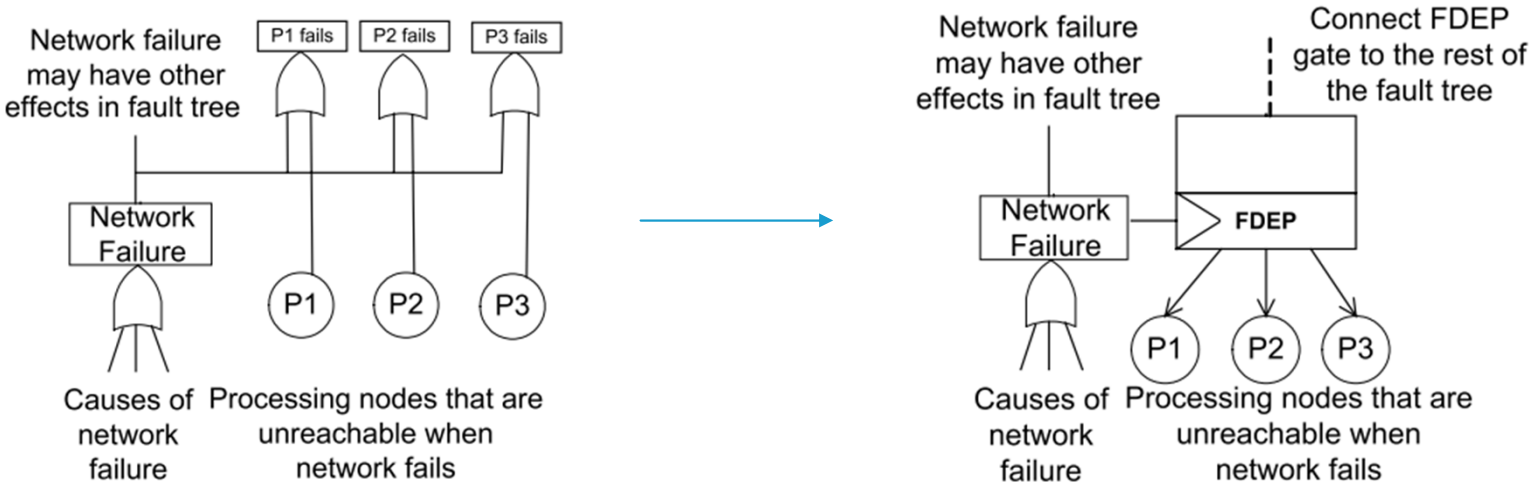
PRIORITY AND GATE (PAND)

- E’ una versione modificata del gate AND, l’output diventa vero se tutti gli eventi in input accadono in uno specifico ordine. Normalmente l’ordine va da sinistra verso destra



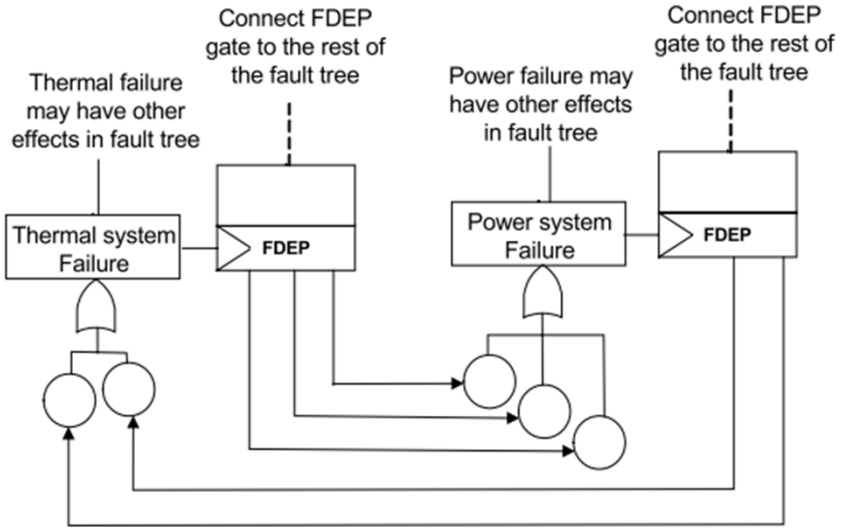
FUNCTIONAL DEPENDENCY GATE (FDEP)

- Supponiamo di avere un sistema configurato in modo tale che il verificarsi di un evento rende inutilizzabili altri componenti. Come ad esempio nel caso del fallimento della rete comporta l'isolamento dei nodi connessi
- Un Functional Depency gate(FDEP) può essere utilizzato per modellare questa condizione. Il gate ha un singolo ingresso denominato trigger input e diversi dependent basic events. Quando l'evento trigger input si verifica i dependent basic event sono forzati a fallire.



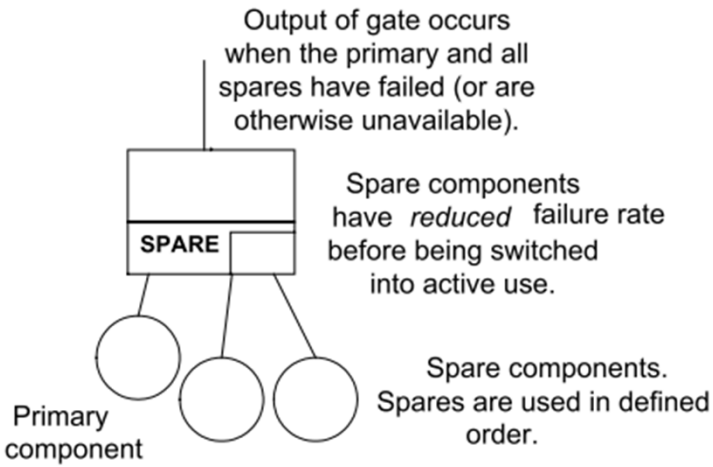
FUNCTIONAL DEPENDENCY GATE (FDEP)

- L’utilizzo dei FPED gate consente di modellare loops quindi interdipendenze, a differenza dei FT classici.
- Supponiamo di avere un sistema di controllo termico ed uno di alimentazione elettrica, entrambi necessitano dell’altro per funzionare



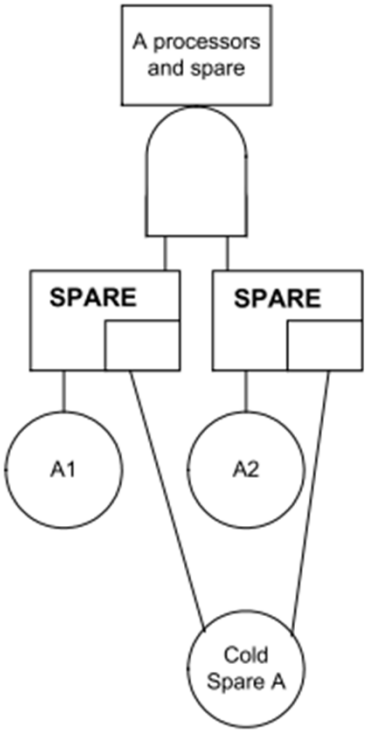
SPARE GATE

- Si consideri un sistema che utilizza i cold spare, ossia componenti che non sono alimentati e quindi non falliscono prima di essere utilizzati.
- Questi sistemi possono essere difficili da modellare con le tecniche standard di FT, perché i criteri di fallimento del sistema non possono essere espressi in termini di combinazioni logiche di eventi di base, tutti con lo stesso intervallo di tempo.
- Il gate Spare viene introdotto per modellare l'attivazione sequenziale degli spares, il primo spare si attiva quando il primario fallisce, cosi via
- Il gate ha un'uscita che diventa vera dopo che si è verificato il fallimento di tutti gli spare component



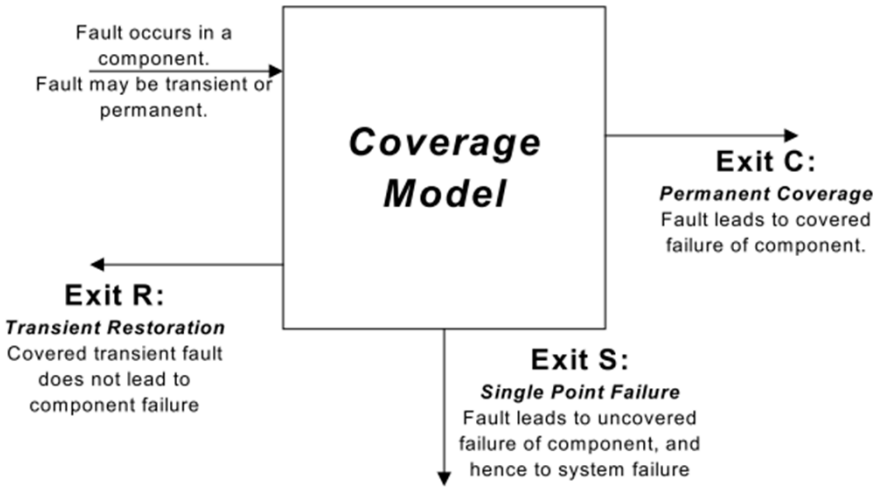
SPARE GATE

- Associato ad ogni ingresso c'è un dormancy factor [0,1], il quale viene moltiplicato per il failure rate. Esso definisce la tipologia dello spare:
  - Cold: non falliscono prima dell'attivazione poiché sono spenti, dormancy 0
  - Hot: possono sempre fallire poiché sono accesi anche se non utilizzati, dormancy 0
  - Warm: sono il caso intermedio
- Gli Spare gate possono anche condividere uno stesso spare, che sarà di ausilio allo Spare gate che fallisce prima



COVERAGE MODEL

- La Coverage è definita come la capacità di un sistema di ripristinare il suo funzionamento dopo un fallimento
- Il modello di coverage rappresenta il verificarsi di un fault e le tre possibili evoluzioni (R,C,S)

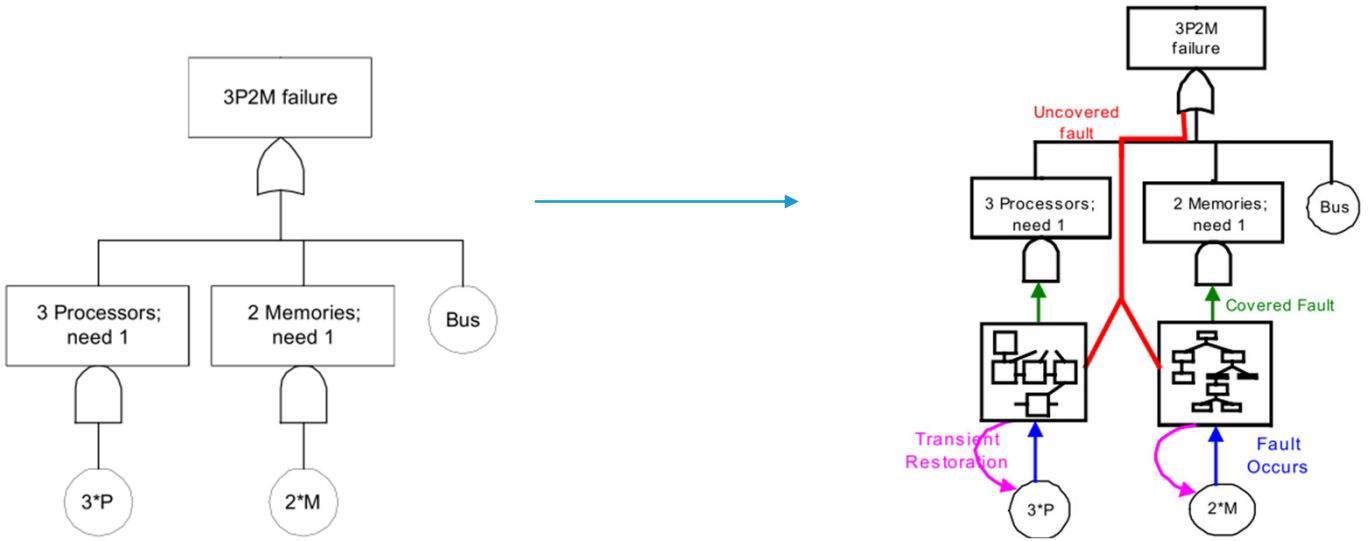


COVERAGE MODEL

- Transient restoration(R), rappresenta il recupero da un errore transitorio, ad esempio mascherando l'errore, ripetendo un'istruzione o tornando a un checkpoint precedente. Le cause sono difetti del software, fattori ambientali. La maggior parte dei fault nei sistemi informatici sono di tipo transient
- Permanent coverage(C), rappresenta la natura permanente del guasto e il successo dell'isolamento e della rimozione (logica) del componente difettoso. Il guasto è persistente: continua ad esistere finché il componente difettoso non viene riparato o sostituito
- Single point of failure(S), rappresenta il caso in cui un singolo guasto causa il malfunzionamento del sistema (l'errore non rilevato si propaga attraverso il sistema o l'unità difettosa non può essere isolata)
- Ad ogni componente si associano tre coverage factors (r,c,s), rappresentano la probabilità di prendere l'exit associata. Rispettivamente
  - s: la probabilità che il fallimento del componente possa far fallire il sistema.
  - r: la probabilità che il sistema possa ripristinarsi automaticamente da un fault transient
  - c: la probabilità che il sistema possa ripristinarsi automaticamente da un fault permanent

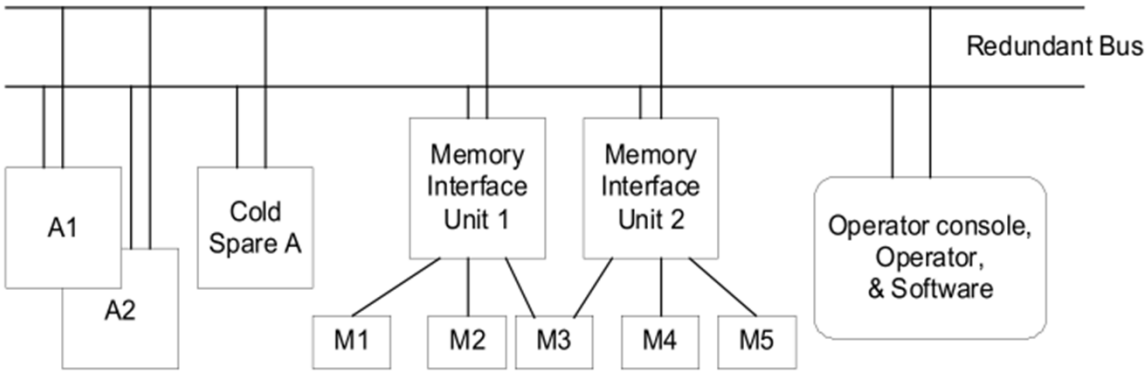
COVERAGE MODEL EXAMPLE

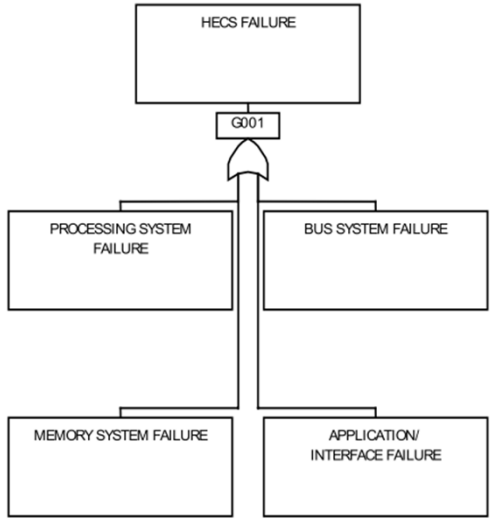
- Il verificarsi di un fault è rappresentato dall'arco dal basic event al coverage model
- I fault di tipo transient, fanno si che il flow ritorni al basic event, dato che non ci sono conseguenze permanenti
- I fault di tipo covered, proseguono verso il gate del FT, poiché non è sicuro che il sistema sia capace di ripristinarsi
- Infine, i fault di tipo uncovered causano il fallimento del sistema, quindi sono messi il OR con il FT



DFTA HECS EXAMPLE

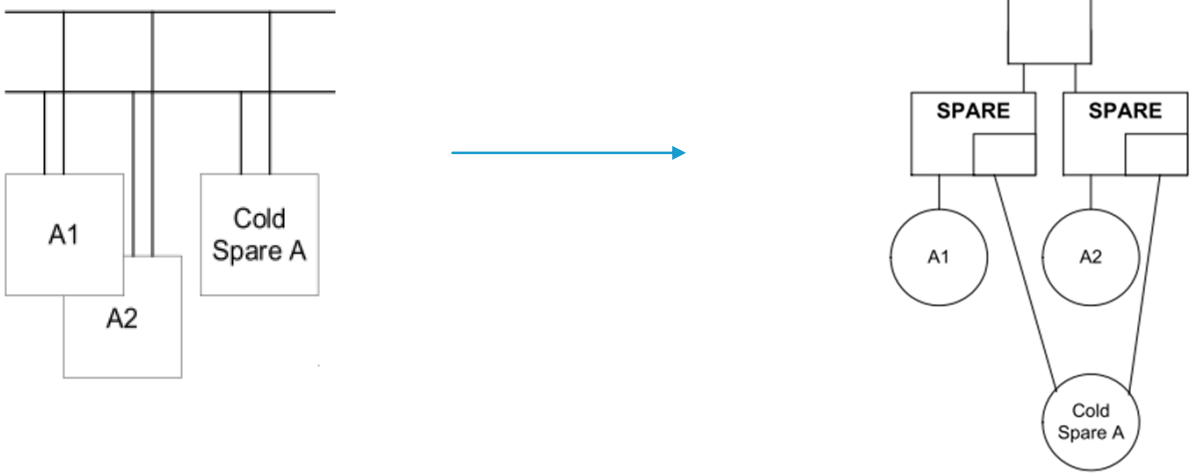
- Hypothetical Computer System(HECS) example
- Il sistema HECS per funzionare necessita del funzionamento di tutti e quattro i sottosistemi: processing, memory, bus e software application.
- Il sistema fallisce se fallisce almeno uno di questi quattro componenti





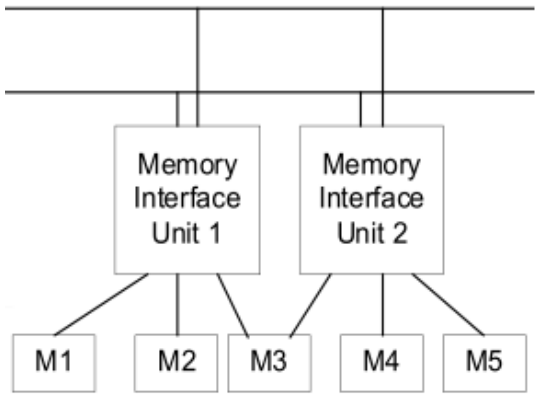
PROCESSING SYSTEM

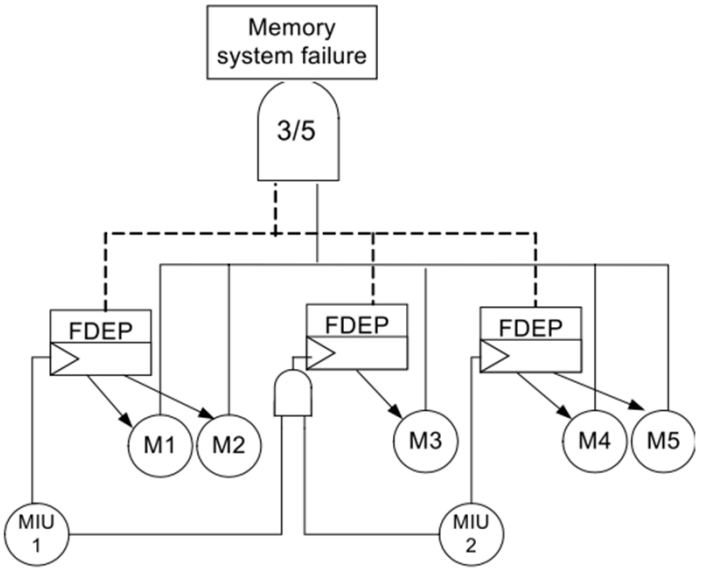
- Il sottosistema è composto da due processori ridondanti A1,A2 che operano in parallelo, un terzo processore configurato come cold spare convdiviso, i tre processori sono identici.
- Quando fallisce A1 oppure A2, il cold spare entra in funzione.
- Il sistema può funzionare finchè non falliscono tutti i tre processori



MEMORY SYSTEM

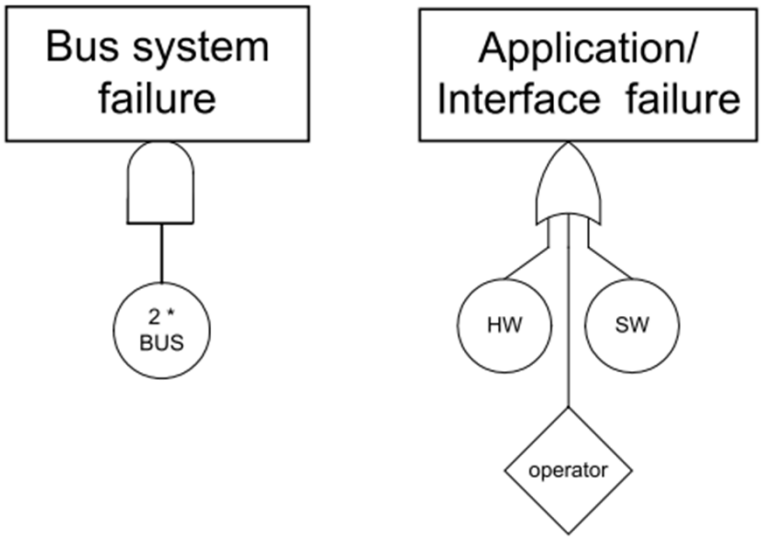
- HECS è composto da cinque unità di memoria, per il funzionamento del sistema ne sono richieste tre
- Le unità di memoria sono connesse al bus tramite due interfacce di memoria ridondanti, le unità di memoria sono funzionalmente dipendenti dalle interfacce
- I gate FDEP non producono output, sono connessi al FT tramite linee fittizie



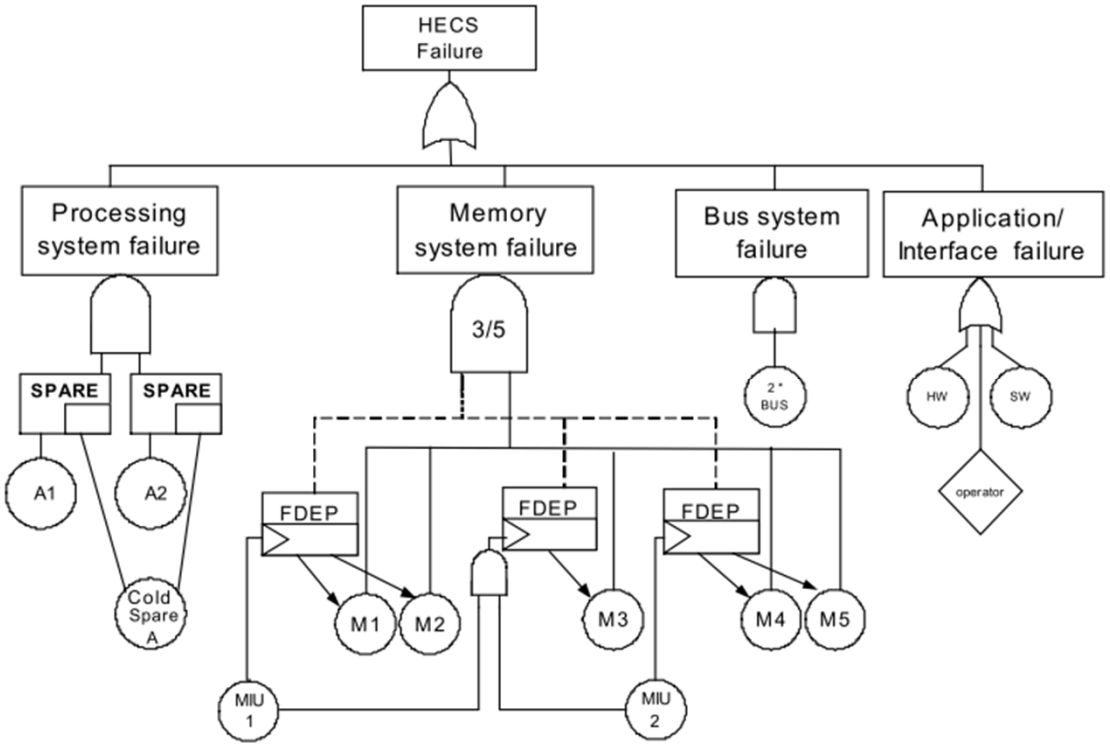


APPLICATION INTERFACE E BUS SYSTEM

- Il sistema software è in esecuzione sul computer di bordo
- L'operatore umano interagisce con il sistema tramite una GUI, in esecuzione su un device esterno
- Il fallimento del sistema può derivare dal fallimento del software, hardware o da un errore umano
- Sono presenti due bus identici ridondanti. Il sistema fallisce se falliscono entrambi i bus



HECS FAULT TREE

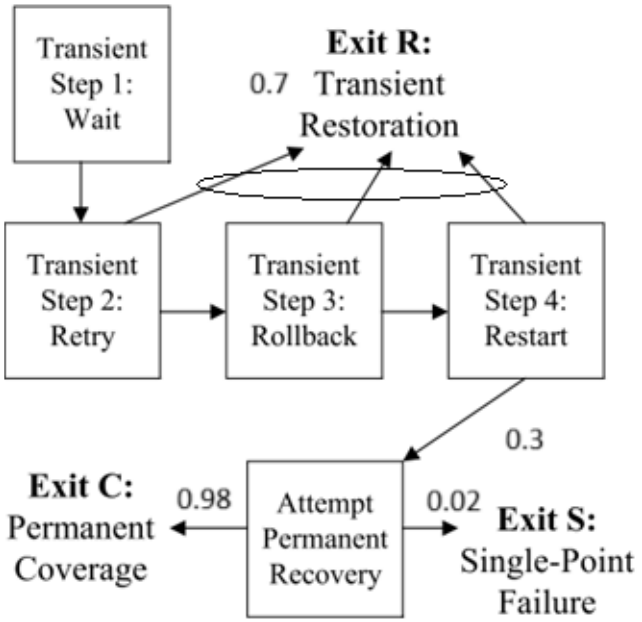


HECS DFT QUANTIFICATION

- Sono necessari due parametri per caratterizzare ogni basic event: failure rates ed i parametri di coverage
- Failure rates
  - Processore $\lambda P = 10^{-4}$ per ora, dato che si assume un cold spare, il dormancy factor è zero
  - Unità di memoria $\lambda M = 6 * 10^{-5}$ per ora
  - Interfacce di memoria $\lambda MIU = 5 * 10^{-5}$
  - Bus $\lambda B = 10^{-6}$ per ora
  - GUI $\lambda HW = 5 * 10^{-5}$ per ora
  - Errore umano Po $= 0.001$
  - Software $P_{SW} = 0.03$

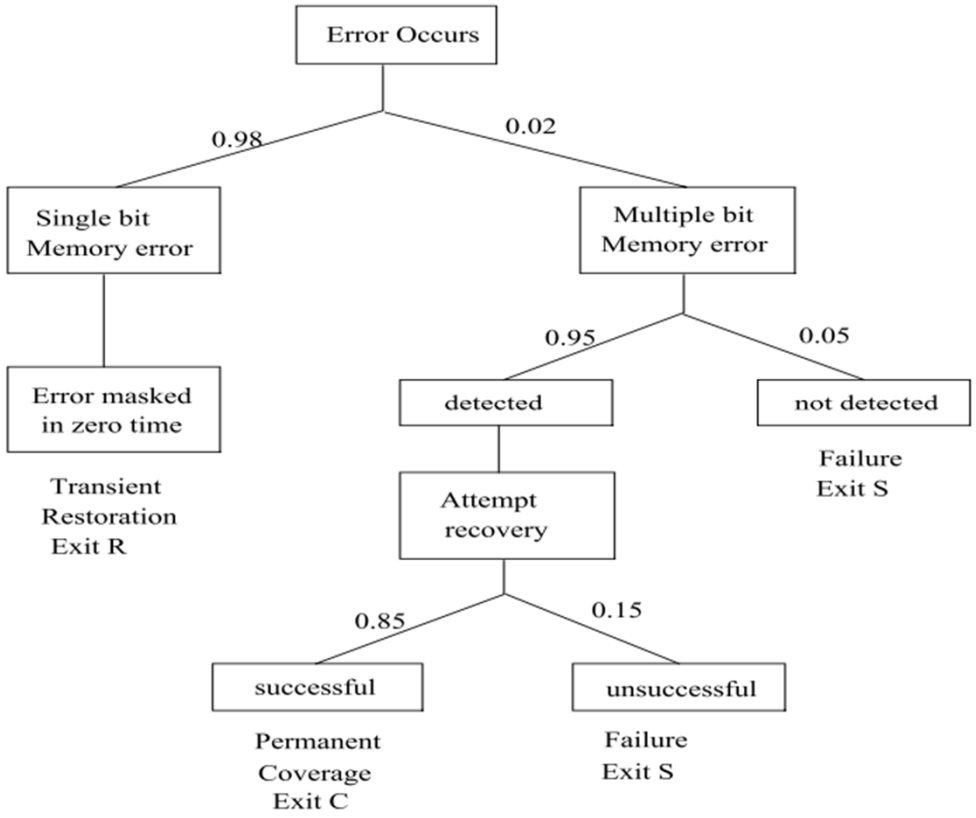
COVERAGE PROCESSORE

- Il processore comprende delle routine built-in che permettono di eseguire un controllo degli errori.
  - Wait attende che il fault si risolva da solo
  - Retry riesegue l'istruzione più volte,
  - Rollback riparte da un checkpoint
  - Restart si esegue se fallisce anche il Rollback
- Nel caso l'errore persiste, si assume essere causato da un permanent fault
- Nel sistema in questione si hanno i seguenti parametri:
  - Rp = 0.7 la procedura di transient restoration sia risolutiva nel 70% dei fault
  - Cp = 0.3 * 0.98 = 0.294, poiché per il 30% degli altri casi, la procedura di permanent coverage ha successo nel 98% dei casi
  - Sp = 1-Cp-Rp = 0.006



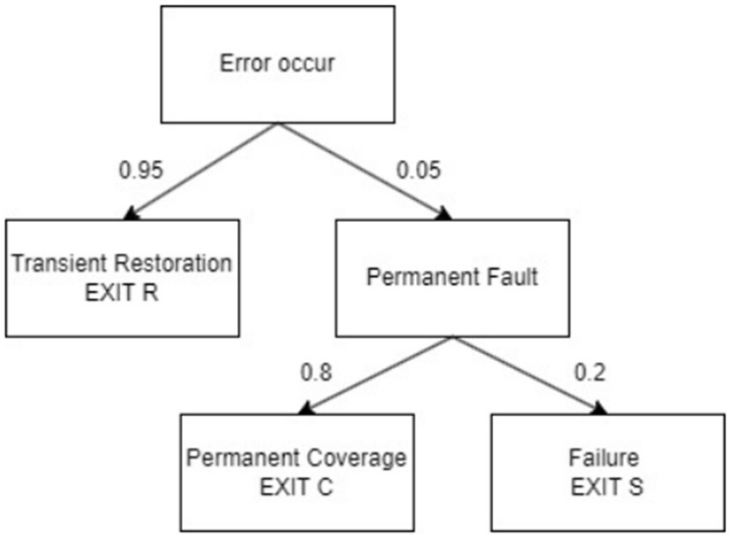
COVERAGE MEMORIA

- La memoria utilizza un codice di correzione degli errori, l'errore sul singolo bit può essere rilevato e corretto.
- Si assume un errore sul singolo bit nel 98% dei casi
  - Rm = 0.98
- Per il 2% quindi si hanno errori su più bit, si assume che nel 95% dei casi siano rilevati e che si possa ripristinare l'85% di essi, otteniamo
  - Cm = 0.02*0.95*0.85 = 0.01615
- Nei casi rimanenti si ottiene un fallimento del sistema
  - Sm = 0.02*(0.05 + 0.95*0.15) = 0.00385



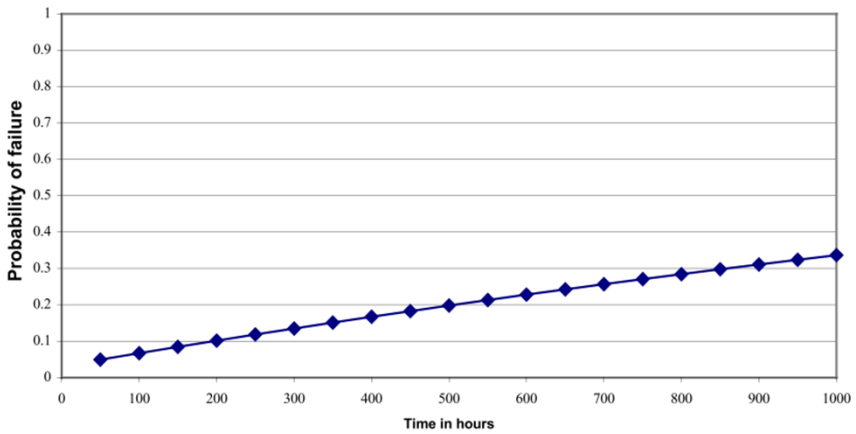
COVERAGE INTERFACCIA MEMORIA, BUS, GUI

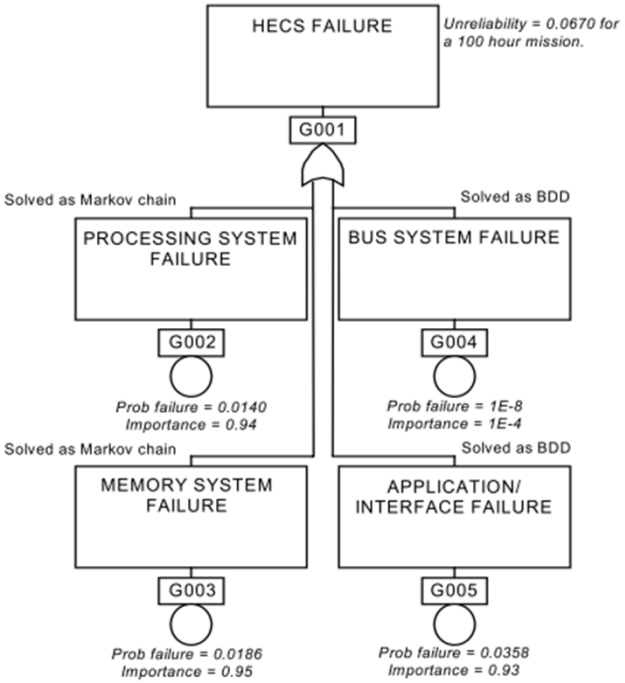
- L'interfaccia di memoria è caratterizzata dal 95% di errori risolvibili con la procedura di transient recovery
  - Rmiu = 0.95
- Il resto del 5% degli errori, è assunto essere risolvibile per l'80% dei casi
  - Cmiu = 0.05*0.80 = 0.04
- Smiu = 1 – Cmiu - Rmiu = 0.01
- Per il resto dei componenti (sw, hw, bus) si assume che i fault siano tutti permanent e risolvibili c = 1, s=r=0



RISULTATI ANALISI

- I sottosistemi di processori, memoria e application interface hanno importanza simile, dato che la Birnbaum importance è circa 1, un piccolo decremento nella unreability in uno di questi sistemi comporta un decremento della stessa portata nella unreability del sistema HECS.
- L'analisi è stata effettuata su un mission time di 100 ore, ottenendo una unreability di circa 0.067, quindi il sistema è circa del 93% affidabile per mission time di 100 ore



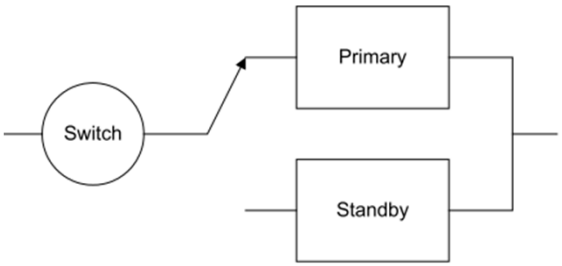


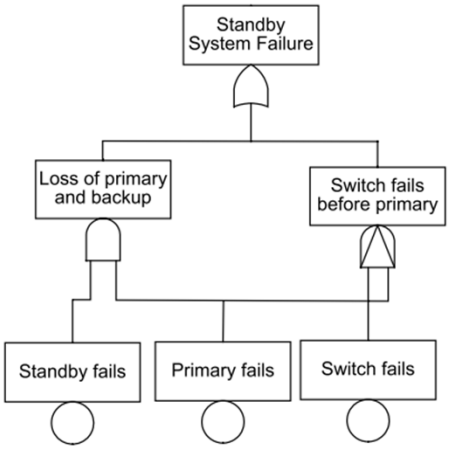
DFT TO MARKOV MODEL

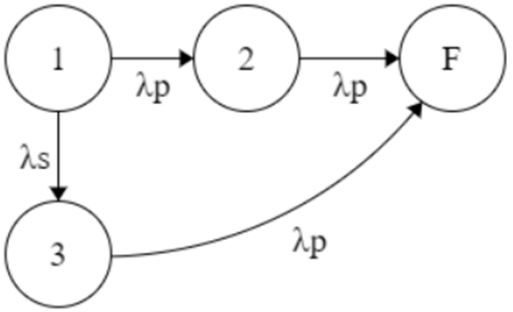
- Le tecniche di analisi per i Fault Tree statici non sono applicabili per i DFT, poiché esse sono basate sui MCSs, ma per i DFT oltre la probabilità con cui si verificano gli eventi, è necessario modellare anche l'ordine in cui essi si verificano
- Le metodologie usate per i DFT sono soluzioni algebriche, Modelli di Markov, Petri Nets, Bayesian Network, Monte Carlo simulation

DFT TO MARKOV MODEL EX

- Sugli archi sono presenti i rate con cui le transizioni avvengono, quindi i failure rate dei componenti. Il componente primario ed il secondario(spare) quando attivo, falliscono con un rate λp mentre λs è il failure rate dello switch.
- Nello stato 1 sono funzionano correttamente tutti i componenti

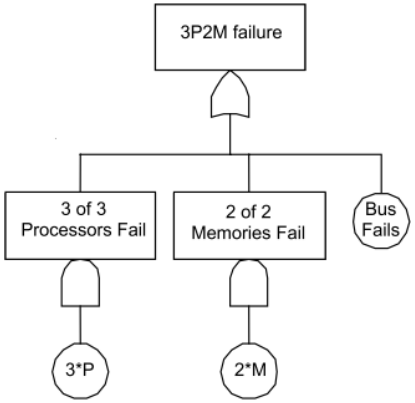


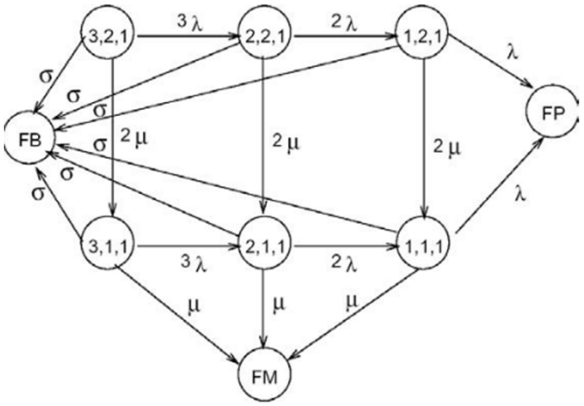




DFT TO MARKOV MODEL EX

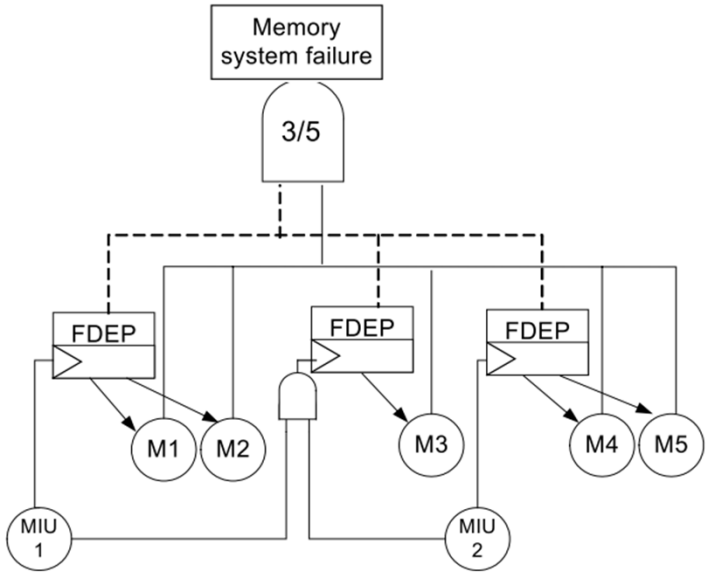
- Failure rate: λ processore, μ memoria, σ bus
- Ogni stato è etichettato con tre valori interi i quali rappresentano il numero di processori, unità di memoria e bus funzionanti. Lo stato iniziale è (3,2,1), si assume che possa fallire un componente per volta
- Esistono due percorsi dallo stato iniziale allo stato (2,1,1), essi rappresentano i differenti ordini con cui il fallimento può avvenire. Nel sistema 3P2M non conta l'ordine in cui falliscono i componenti(solo AND,OR), se fosse stato presente un comportamento dinamico lo stato (2,1,1) sarebbe stato suddiviso in due sottostati per modellare la sequenza con cui vi si arriva

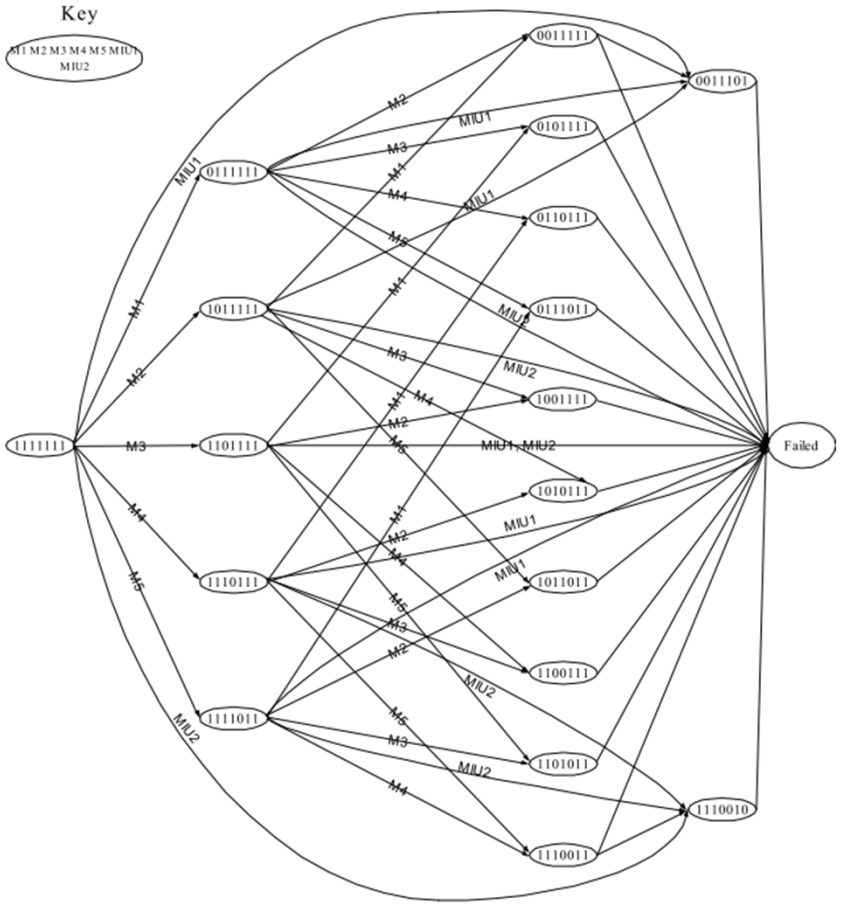




HECS MEMORY MARKOV MODEL

- Il fallimento di una unità di memoria Mi, porta da uno stato i ad uno con i+1 fallimenti. Il fallimento di una interfaccia di memoria ad uno stato con i+3 fallimenti, poiché le unità di memoria collegate non sono più accessibili
- Gli archi senza etichetta, indicano che qualsiasi evento di fallimento fa transitare



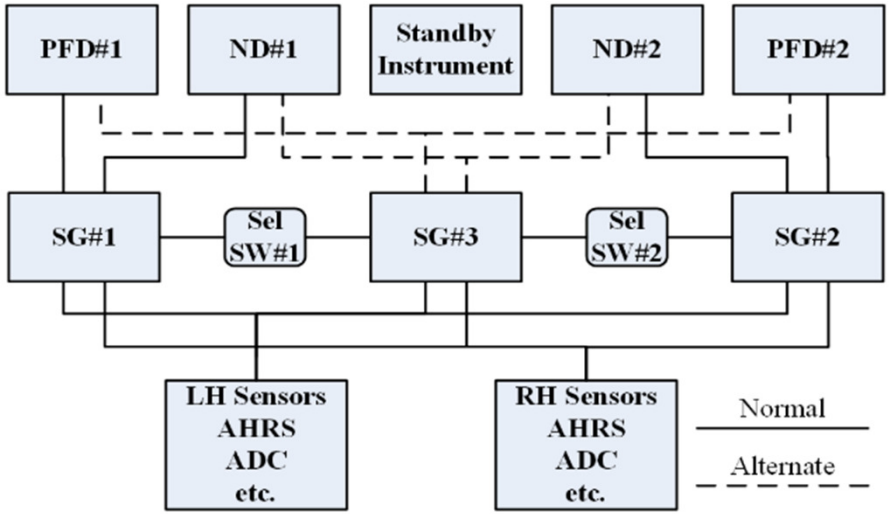


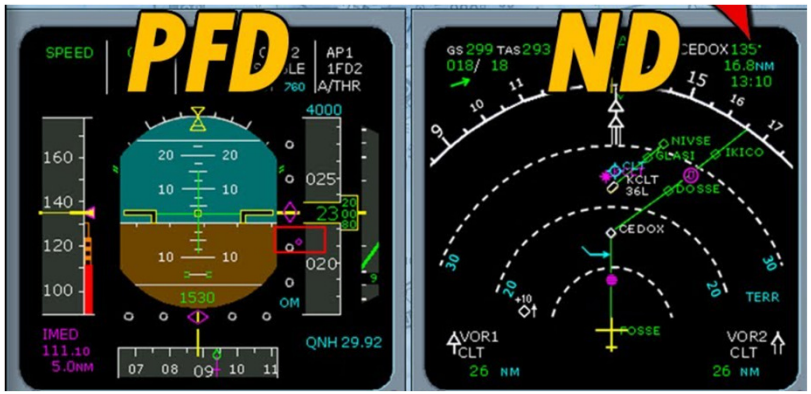
DFT EFIS EXAMPLE

- Electronic Flight Instrument System (EFIS) è un deck display che fornisce informazioni sul volo e sulla navigazione in ambito aereonautico. Per migliorare l'affidabilità sono utilizzate tecniche di ridondanza.
- Vi sono due tipologie di display Primary Flight Display(PFD) ed un Navigation Display(ND), sono utilizzati per visualizzare dati differenti. Essendoci due piloti, sono presenti quattro schermi in totale. Lato 1: PFD1,ND1, Lato 2: PFD2,ND2
- Sono presenti tre Symbol Generator(SG) i quali leggono i dati dai sensori e producono il video sui display collegati. Ogni SG può produrre simultaneamente sia dati per il PFD che per il ND.
- Normalmente SG1 produce il video per PFD1 e ND1, SG2 per PFD2 e ND2. Inoltre è presente un generatore ausiliare SG3 hot spare.
- Nel caso ci fosse un guasto ad uno dei display, ogni lato ha uno switch SW1, SW2 tramite il quale è possibile visualizzare sul display funzionante dello stesso lato entrambe le tipologie di informazioni PFD&ND
- Infine, è presente un sistema indipendente in standby (ISIS) da utilizzare nel caso di fallimento del deck display

DFT EFIS EXAMPLE

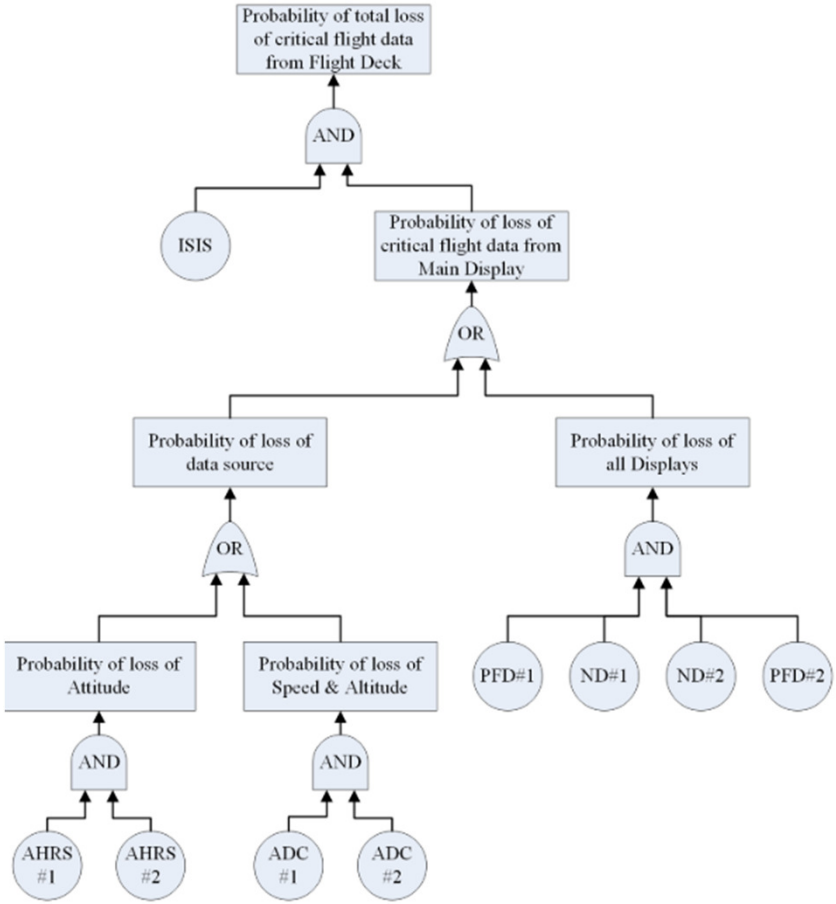
- La US and European Aviation Certification Authorities classifica l'evento di «total loss of critical flight data from the flight deck» come fallimento catastrofico, la sua probabilità di occorrenza deve essere inferiore a 10^{-9} per ora di volo





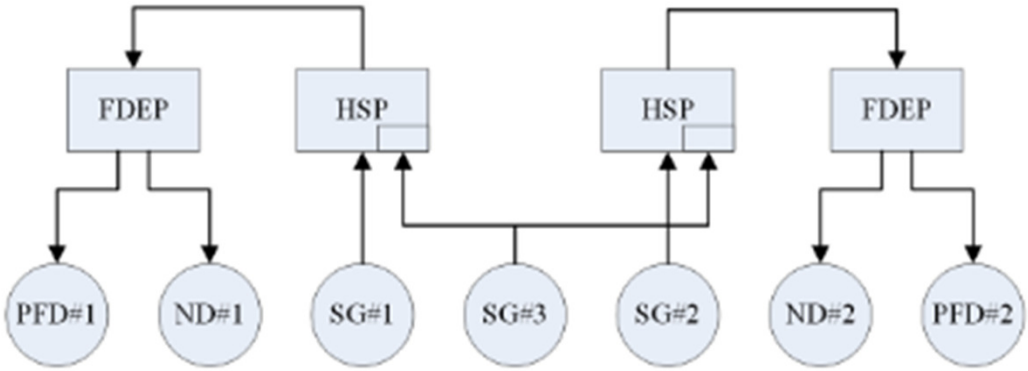
MAIN SYSTEM

- EFIS è composto da due sottosistemi indipendenti ISIS ed il main display deck, sono messi in AND
- Il sistema main display deck è composto da un sottosistema di acquisizione dati, quattro display.
- Il sottosistema di acquisizione dati, è composto da due tipologie di sensori che forniscono diverse informazioni, entrambi ridondati. Il fallimento di uno delle due tipologie di sensori comporta il fallimento del sottosistema, dato che entrambe le tipologie di informazioni sono cruciali
- Sono presenti quattro display indipendenti, due per lato. Dato che ogni display può lavorare in modalità mista PDF&ND, i display di ogni lato sono in AND(filo comune). Il fallimento del sottosistema è causata dal fallimento di tutti i display di entrambi i lati



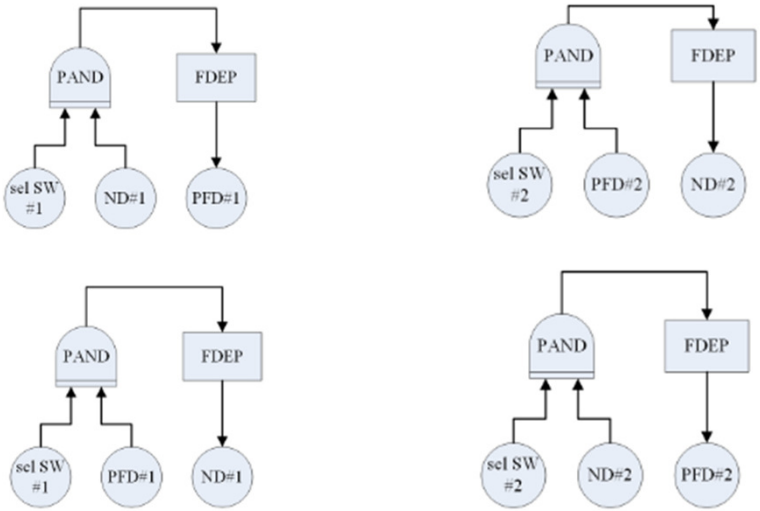
SYMBOL GENERATOR SYSTEM

- Il display FD1 e ND1 sono alimentati da SG1, analogamente FD2 e ND2 sono gestiti da SG2. Per entrambi i sistemi il modulo SG3 può essere utilizzato come sistema ausiliare.
- Il fallimento di SG1 e SG3 rende inutilizzabili FD1 e ND1, mentre il fallimento di SG2 e SG3 compromette FD2 e ND2.
- E' possibile modellare le dipendenze tra i componenti tramite il gate FDEP, inoltre il modulo SG3 ausiliare può essere modellato come hot spare HSP



DISPLAY FAILURE

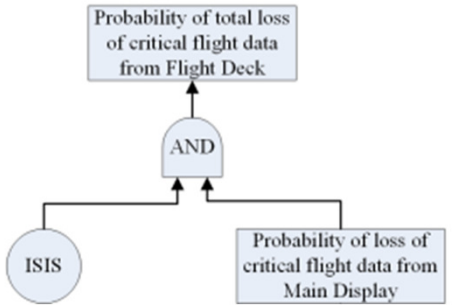
- Nel caso lo switch di un lato fallisca prima del modulo PFD o ND, l'altro display non potrà essere utilizzato come display integrato PFD&ND, è assunto essere inutilizzabile. Questa condizione è modellata tramite il gate PAND e FDEP. Di seguito sono riportati due diagrammi, uno per lato



RISULTATI

- Si assume che i componenti siano non riparabili, i fallimenti indipendenti, failure rate costanti e che i componenti falliscano uno per volta
- La probabilità del top event «total loss of critical flight data from the flight deck» deve essere inferiore a 10^{-9} per ora di volo
- Dall'analisi del sistema si ottiene «loss of probability of critical flight data on the main display" pari a $6.94989 * 10^{-9}$ per un ora di volo
- La probabilità del top event si ottiene mettendo in AND il sottosistema ISIS, ottenendo
  - $6.94989 * 10^{-9} * 2.0 * 10^{-4} = 1.38998 * 10^{-12} < 10^{-9}$
  - Il sistema rispetta le specifiche di safety richieste

Component	MTBF	Failure Rate per hour
Integrated Standby Instrument System (ISIS)	5000 hrs	2.0×10^{-4}
Attitude and Heading Reference System (AHRS)	15000 hrs	6.67×10^{-5}
Air Data Computer (ADC)	20000 hrs	5.0×10^{-5}
Display Unit (DU)	7500 hrs	1.33×10^{-4}
Symbol Generator (SG)	10000 hrs	1.0×10^{-4}
Select Switch (Sel SW)	30000 hrs	3.33×10^{-5}



REFERENCES

- W. E. Vesely et al. "Fault Tree Handbook with Aerospace Applications. 2002." NASA Office of Safety and Mission Assurance.
- D. Haiyong, W. Guoqing, Z. Zhengjun, L. Yanhong and G. Qingfan, "Availability analysis of electronic flight instrument system based on dynamic fault tree," 2018 Tenth International Conference on Advanced Computational Intelligence (ICACI), Xiamen, China, 2018, pp. 461-466, doi: 10.1109/ICACI.2018.8377503.