MINISTERO DELLO SVILUPPO ECONOMICO
DIREZIONE GENERALE PER LA TUTELA DELLA PROPRIETA' INDUSTRIALE
UFFICIO ITALIANO BREVETTI E MARCHI

UIBM

DOMANDA NUMERO	**102014902306068**
Data Deposito	**31/10/2014**
Data Pubblicazione	**01/05/2016**

Sezione	Classe	Sottoclasse	Gruppo	Sottogruppo
G	06	F		

Sezione	Classe	Sottoclasse	Gruppo	Sottogruppo
G	06	F		

Titolo

PROCEDIMENTO PER MISURARE L'EFFETTO DI GUASTI HARDWARE MICROSCOPICI IN APPLICAZIONI AD ELEVATA COMPLESSITA IMPLEMENTATE IN UN SISTEMA ELETTRONICO HARDWARE, RELATIVO SISTEMA E PRODOTTO INFORMATICO

DESCRIZIONE dell'invenzione industriale dal titolo:

"Procedimento per misurare l'effetto di guasti hardware microscopici in applicazioni ad elevata complessità implementate in un sistema elettronico hardware, relativo sistema e prodotto informatico"

di: Yogitech S.p.A., nazionalità Italiana, Via Lenin 132/p loc. S. Martino Ulmiano 56017 S. Giuliano Terme PI, Italia.

Inventori designati: Riccardo MARIANI

Depositata il: 31 ottobre 2014

* * *

TESTO DELLA DESCRIZIONE

Campo tecnico

La presente descrizione si riferisce a tecniche per misurare l'effetto di guasti hardware microscopici in applicazioni ad elevata complessità implementate in un sistema elettronico hardware, comprendente originare cause di guasto hardware a livello microscopico e misurare un corrispondente effetto finale in detta applicazione a elevata complessità. Varie forme di attuazione possono essere applicate alla sicurezza funzionale, in particolare in sistemi elettronici nel campo della robotica industriale e dei controlli industriali, e nel campo tecnico dei sistemi elettronici per applicazioni automobilistiche di assistenza al conducente e guida parzialmente o completamente automatica.

Sfondo tecnologico

Le norme per la sicurezza funzionale come la norma IEC 61508 e la norma ISO 26262 richiedono, per sistemi elettronici utilizzati in applicazioni critiche per la vita, di verificare la capacità di tali sistemi di rilevare o tollerare guasti hardware od errori software.

Con il crescere dell'utilizzo dei sistemi elettronici, tali applicazioni critiche sono sempre più complesse, per complessità in questo caso intendendosi la complessità delle funzioni software che vengono eseguite. Tale complessità può essere ad esempio misurata tramite la cosiddetta 'complessità ciclomatica' sviluppata da Thomas J. McCabe nel 1976.

Un esempio di tali applicazioni critiche ad elevata complessità è il veicolo a guida autonoma, ossia autoveicoli che sono in grado di muoversi senza o con limitato intervento umano. E' evidente sia che un guasto in un tale veicolo può avere conseguenze molto gravi per il passeggero o per i pedoni, ma anche che tale applicazione è molto complessa perché composta da molte funzioni software strettamente interfacciate tra di loro (rilevazione della corsia, della distanza dai veicoli circostanti, lettura dei segnali stradali e altre).

Come detto, le norme per la sicurezza funzionale si preoccupano sia di guasti hardware che di errori software. Le soluzioni qui discusse si riferiscono specificamente ai guasti hardware.

Uno dei metodi indicati dalle stesse norme sopra menzionate per verificare la capacità del sistema di rilevare o tollerare guasti hardware è la cosiddetta "iniezione guasti", ossia una forma di verifica che prevede l'iniezione intenzionale di malfunzionamenti al fine di constatare che tali guasti non hanno effetto sull'applicazione oppure sono rilevati da opportuni sistemi di controllo.

Tecniche di iniezioni guasti hardware sono note allo stato dell'arte ma presentano due principali inconvenienti.

Quando le iniezioni di guasti hardware sono effettuate a livello di test di un sistema elettronico fisico in laboratorio, tali tecniche sono in grado di misurare l'effetto di guasti hardware macroscopici come ad esempio mancata alimentazione o rottura di un collegamento a livello di circuito stampato, ma non sono assolutamente in grado di misurare l'effetto di guasti microscopici come ad esempio l'effetto di un guasto su un transistor di uno dei circuiti integrati facenti parte del sistema elettronico, perché a livello di laboratorio non è possibile iniettare tali guasti all'interno del circuito integrato se non con operazioni costosissime (quali ad esempio l'irradiazione del componente a particella alfa).

Quando le iniezioni di guasti hardware sono effettuate a livello di simulazione hardware del sistema elettronico, come indicato ad esempio nel brevetto statunitense US7472051 B2 a nome della Richiedente, tali tecniche sono in grado di misurare l'effetto locale di un guasto hardware, ossia ad esempio come un circuito integrato reagisca ai suoi morsetti in caso di un guasto microscopico quale la rottura di un transistor interno al circuito integrato. Tuttavia, tali tecniche diventano inefficaci quando si voglia misurare l'effetto finale, ossia l'effetto che il guasto microscopico ha sull'applicazione finale. Ad esempio nel veicolo a guida autonoma, l'effetto finale d'interesse causato da un guasto microscopico può essere l'attuazione del sistema frenante quando non necessario, causata da un'erronea rilevazione di un ostacolo. Si può immaginare come sia inattuabile misurare l'effetto finale a partire dall'iniezione di un guasto microscopico, ad esempio per la durata del tempo di simulazione, perché la distanza (intesa come la catena di eventi tra causa e

effetto) tra la causa iniziale (ossia il guasto fisico microscopico) e l'effetto macroscopico finale (ossia l'erronea rilevazione di un ostacolo) è troppo grande.

A questo riguardo, in figura 1 sono mostrati schematicamente dei modi di fallimento, o failure mode, che si identificano nell'ambito delle procedure per misurare l'effetto di guasti hardware microscopici in applicazioni ad elevata complessità implementate in un sistema elettronico hardware, ossia la verifica della sicurezza funzionale dei sistemi elettronici per applicazioni critiche. Tali modi di fallimento corrispondono a comportamenti non corretti del dispositivo o sistema elettronico rispetto ai suoi comportamenti da specifica. Tali modi di fallimento relativi al dispositivo o sistema, detti effetti finali, o modi di fallimento finali FFM, sono rappresentati, ad esempio in campo automobilistico, da eventi come la "l'attuazione del sistema frenante quando non necessario". Come si vede in figura 1, se si scompone il sistema elettronico nei suoi sottosistemi e componenti, tali effetti finali FFM possono essere ricondotti tramite semplici operazioni logiche rappresentate da porte logiche PL, ad esempio porte AND o OR) attraverso livelli di altri modi di fallimento risultanti, SFM, altri modi di fallimento "radice", RFM, ossia legati in modo più diretto ai componenti del sistema. Ad esempio, nel caso di un circuito integrato, un modo di fallimento radice RFM1 può essere "calcolo errato da parte del processore" che in OR logico con un altro fallimento radice RFM2 "indirizzo errato da parte del processore nell'accesso alla memoria" può costituire un modo di fallimento di livello più alto (modo di fallimento "risultante" SFM) quale ad esempio funzionamento errato del processore. Combinando uno o più

livelli di modi di fallimento risultanti si giunge infine modo di fallimento finale FFM, che, come è detto, è quindi una combinazione logica di modi di fallimento radice RFM. In tal senso si intende che la catena di eventi tra causa microscopica e effetto finale può essere molto lunga, ossia il numero di livelli che lega ad esempio il modo di fallimento radice RFM1 all'effetto finale FFM può essere molto elevato.

Nei sistemi per applicazioni critiche, come richiesto dalle normative internazionali IEC 61508 e ISO 26262, si procede alla realizzazione di meccanismi di sicurezza capaci di prevenire o rilevare tali modi di fallimento. Ad esempio nel caso di circuiti integrati, un meccanismo di sicurezza è un software eseguito periodicamente durante il normale funzionamento del dispositivo per testare tutte le istruzioni di un processore.

Le normative internazionali IEC 61508 e ISO 26262 richiedono che tali meccanismi di sicurezza raggiungano certi valori di "copertura diagnostica", ossia un certo rapporto tra il numero dei guasti pericolosi (ossia in grado di perturbare la missione critica del sistema) ed il numero di guasti rilevati dal meccanismo diagnostico.

In questo caso, lo scopo della verifica della sicurezza funzionale è quello di verificare che – dato l'insieme dei guasti hardware microscopici possibili – il numero di guasti pericolosi e il numero di guasti rilevanti dal meccanismo diagnostico siano quelli preventivati in fase di progetto del sistema elettronico.

In figura 2 sono illustrati schematicamente elementi e passi di misurare dell'effetto di guasti hardware microscopici in applicazioni ad elevata complessità implementate in un sistema elettronico hardware, ad esempio

un circuito integrato, indicato con il riferimento numerico 11, sottoposto a verifica. In figura 2 è rappresentato un guasto microscopico G iniettato a una locazione LC, nel circuito integrato 11 sotto verifica. Con x, y,…,z è indicata una pluralità di uscite designate del circuito integrato 11 sotto verifica. Una funzione f(x,y,…,z) di tali uscite designate x, y,…,z rappresenta un punto di osservazione O del guasto MG. Una seconda funzione G (x,y,…,z) delle uscite designate x, y,…,z rappresenta un punto diagnostico D. Naturalmente tali grandezze e le operazioni e funzioni connesse sono definite per un fase di simulazione del circuito integrato 11 sotto verifica, ad esempio tramite un elaboratore o computer 600 che può caricare in memoria ed esegue un programma informatico 610, in particolare su un supporto informatico, o un mezzo non transitorio leggibile da elaboratore, che comprende una o più delle operazioni del procedimento qui descritto, anche se in generale tali operazioni potrebbero essere eseguite sul circuito fisico, ancorché con le difficoltà discusse in precedenza.

Tale misura degli effetti di guasti hardware può essere inglobata in una fase di simulazione del sistema elettronico o circuito elettronico 11 con iniezione di guasto, che, tipicamente, comprende:

iniettare dei guasti microscopici MG in fase di simulazione (ad esempio guasti di tipo stuck-at) del circuito 11 in determinate locazioni LC di tale circuito 11;

verificare se una certa funzione f(x,y,…,z) delle uscite designate x, y,…,z, o punto di osservazione O viene perturbata dal guasto, ossia se il punto di osservazione O ha un valore misurato diverso da un valore atteso. In tal

caso il guasto G viene definito come guasto potenzialmente pericoloso, in caso contrario è un guasto sicuro, ossia non causa un fallimento della missione critica. Le uscite designate x,y,…,z sono le uscite del circuito integrato la cui combinazione può appunto determinare il modo di fallimento radice RFM. Ad esempio, il modo di fallimento radice "calcolo errato da parte del processore" ha come uscite designate le uscite del processore;

nel caso di guasto (potenzialmente) pericoloso verificare se la seconda funzione G(x,y...z) delle uscite corrispondenti al meccanismo di sicurezza ("punto diagnostico", D) viene attivata (ossia, ad esempio, il valore di G(x,y...z) differisce da un valore precalcolato) dal guasto pericoloso ossia se, quando il punto di osservazione O assume un valore misurato diverso da quello atteso , il punto diagnostico D viene attivato entro un certo intervallo di tempo individuato dalle specifiche di sicurezza del dispositivo elettronico 11. Se il punto diagnostico D viene attivato, si intende una condizione di guasto pericoloso rilevato. Ad esempio, nel caso che il meccanismo di sicurezza sia un software eseguito periodicamente, il punto diagnostico D è rappresentato dal registro del processore o dalla locazione di memoria in cui il software deposita il suo valore misurato, da confrontarsi con un valore pre-calcolato del test (ossia il valore atteso senza guasti). Usualmente, l'iniezione di guasti avviene mentre un generatore di carico di lavoro fa eseguire comandi, in particolare ad esempio un applicazione in sicurezza funzionale, al sistema elettronico. Sono previsti moduli di monitoraggio per tracciare l'esecuzione dei comandi e raccogliere i dati, ad esempio dai detti punti di osservazione, e moduli di analisi, che eseguono ad

esempio le valutazioni dei guasti pericolosi, ad esempio secondo l'analisi FMEA. Tutti tali moduli possono essere moduli software eseguiti su un elaboratore che più in generale esegue la procedura di simulazione. Simili procedure sono descritte ad esempio nel brevetto US7937679 a nome della Richiedente.

Scopo e sintesi

Le forme di attuazione qui descritte hanno lo scopo di migliorare le potenzialità dei procedimenti secondo la tecnica nota come discussi in precedenza, in particolare riducendo la distanza tra la potenziale causa microscopica iniziale e la misura dell'effetto finale e quindi superare i due inconvenienti sopra descritti, ossia permettere velocità di misura molte elevate e permettere operazioni di modifica per originare tali cause a livello di guasti microscopici.

Varie forme di attuazione raggiungono tale scopo grazie ad un procedimento avente le caratteristiche richiamate nelle rivendicazioni che seguono. Varie forme di attuazione possono riferirsi anche a un sistema di simulazione così come possono riferirsi ad un prodotto informatico, caricabile nella memoria di almeno un computer (ad es. un terminale in una rete) e comprendente porzioni di codice software adattate per eseguire i passi del procedimento nel momento in cui il programma viene eseguito su almeno un computer. Come qui utilizzato, tale prodotto informatico è inteso essere equivalente ad un mezzo leggibile da computer contenente istruzioni per controllare il sistema informatico in modo da coordinare l'esecuzione del procedimento secondo l'invenzione. Il riferimento ad "almeno un computer" è inteso sottolineare la possibilità

per la presente invenzione di essere implementata in una forma modulare e/o distribuita. Le rivendicazioni formano una parte integrale degli insegnamenti tecnici qui somministrati in relazione all'invenzione.

Varie forme di attuazione possono prevedere che il procedimento comprenda un'operazione di iniettare guasti comprendente selezionare un guasto microscopico da iniettare, selezionare in una libreria di mutanti software un mutante corrispondente a detto guasto microscopico da iniettare, applicare detto mutante selezionato a detta istanza software per ottenere un istanza mutata, simulare detto sistema o circuito elettronico eseguente detta istanza mutata in un determinato scenario di test, misurare l'effetto corrispondente all'istanza mutata.

Varie forme di attuazione possono prevedere di selezionare o creare, in funzione del guasto microscopico da iniettare, una sequenza di scenari comprendenti parametri di sollecitazione dell'istanza dell'applicazione critica, le operazioni di simulare e misurare essendo eseguite una pluralità di volte secondo la sequenza (NST) di scenari di test .

Varie forme di attuazione possono prevedere di selezionare un guasto microscopico da iniettare tramite l'eseguire un'analisi del sistema elettronico per identificare un insieme di guasti hardware microscopici da iniettare, individuare in tale insieme di guasti hardware microscopici dei guasti hardware microscopici più probabili; mentre l'operazione di selezionare un mutante software corrispondente al guasto microscopico da iniettare comprende costruire una libreria (di mutanti software, in particolare organizzata per tipologia, persistenza e quantità di mutanti da applicare); selezionare da detta

libreria dei mutanti (corrispondenti ai potenziali guasti hardware microscopici più probabili).

Varie forme di attuazione possono prevedere di misurare l'effetto corrispondente all'istanza mutata rilevando l'effetto finale e classificandolo secondo classi di effetto finale, in particolare comprendenti classi di evento atteso , evento di sicurezza ed evento errato.

Varie forme di attuazione possono prevedere di operare un passo di calcolo di indici di sicurezza del guasto microscopico in funzione di risultati detto passo di classificazione (280), tramite:

calcolare il numero di eventi attesi, eventi di sicurezza ed eventi errati per una sequenza di test effettuata con un determinato mutante;

calcolare in funzione di detti numero di eventi attesi, eventi di sicurezza ed eventi errati fattori di sicurezza e copertura diagnostica e associare detti fattori a detto guasto microscopico selezionato.

La soluzione descritta rende possibile in modo estremamente semplice ed automatico il collegamento tra guasti hardware microscopici e i loro effetti finali. L'utilizzo di mutazioni software permette di eseguire l'iniezione guasti su un modello software che quindi é, anche per applicazioni estremamente complesse, sia estremamente veloce sia capace di modellare, tramite le mutazioni, gli effetti causati da tali guasti hardware microscopici.

Breve descrizione delle figure

Varie forme di attuazione saranno ora descritte, a puro titolo di esempio, con riferimento alle figure annesse, in cui:

- le Figura 1, 2 sono state descritte in precedenza;

- la Figura 3 mostra un schema generale di sistema implementante il procedimento descritto;

- la Figura 4 mostra uno schema a blocchi rappresentativo di un procedimento per misurare l'effetto di guasti hardware microscopici qui descritto;

- la Figura 5 mostra un diagramma rappresentativo di dettaglio di un'operazione del procedimento di Figura 4;

- la Figura 6 mostra uno schema a blocchi di un'applicazione ad alta complessità verificabile tramite il procedimento qui descritto;;

- la Figura 7 rappresenta un esempio di applicazione del procedimento qui descritto all'applicazione di Figura 6;

- la Figura 8 mostra un diagramma rappresentativo di dettaglio di ulteriori operazioni del procedimento di Figura 4.

Descrizione dettagliata

Nella descrizione che segue vengono forniti numerosi dettagli specifici al fine di consentire la massima comprensione delle forme di attuazione esemplificative. Le forme di attuazione possono essere messe in pratica con o senza dettagli specifici, oppure con altri procedimenti, componenti, materiali, etc. In altre circostanze, strutture materiali od operazioni ben noti non sono mostrati o descritti in dettaglio per evitare di mettere in ombra aspetti delle forme di attuazione. Il riferimento nel corso di questa descrizione ad "una forma di attuazione" significa che una particolare peculiarità, struttura o caratteristica descritta in connessione con la forma di attuazione è compresa in almeno una forma di attuazione. Dunque, il ricorrere della frase "in una forma di attuazione" in vari punti nel corso di questa descrizione

non è necessariamente riferito alla stessa forma di attuazione. Inoltre, le particolari peculiarità, strutture o caratteristiche possono essere combinate in un qualunque modo conveniente in una o più forme di attuazione.

Le intestazioni ed i riferimenti sono qui forniti solo per convenienza del lettore e non definiscono la portata od il significato delle forme di attuazione.

In breve, il procedimento descritto prevede di operare la definizione di un procedimento di misura basato sull'esecuzione di un modello software (programma) dell'applicazione critica ad elevata complessità, modificata tramite mutazioni software, ossia modifiche intenzionali del programma per emulare l'effetto di un errore, mirate da una analisi dettagliata dei possibili guasti hardware microscopici.

Tale procedimento è capace di ridurre la distanza tra la potenziale causa microscopica iniziale e la misura dell'effetto finale e quindi permettere velocità di misura molte elevate, perché la misura avviene a livello di esecuzione di un programma software e non a livello di simulazione di un circuito integrato, e allo stesso tempo permettere l'iniezione di mutazioni corrispondenti a guasti microscopici, e quindi non limitare l'iniezione a soli guasti macroscopici.

Le tecniche di mutazioni del programma sono note di per sé al tecnico del settore, ad esempio dal documento US 8468503 B2, ma sono utilizzate per il test del programma rispetto a potenziali errori software.

La soluzione qui descritta impiega le mutazioni per modellare l'errore nel programma causato da un guasto hardware a livello microscopico e misurare un corrispondente effetto finale nell'applicazione critica a

elevata complessità; in particolare la soluzione qui descritta prevede l'utilizzo della mutazione software come strumento per collegare l'effetto locale di potenziali guasti hardware microscopici con l'effetto macroscopico a livello dell'applicazione, passando per un livello intermedio ossia la mutazione del programma in modo da emulare l'effetto dei guasti hardware microscopici più probabili.

In figura 3 è mostrato schematicamente un sistema che implementa il procedimento qui descritto per misurare l'effetto di guasti hardware microscopici in applicazioni ad elevata complessità. Analogamente al sistema descritto in Figura 2 è mostrato un sistema di elaborazione 600 che opera una fase di simulazione del sistema o circuito elettronico 11. Tuttavia, in questo caso, in luogo di eseguire una istanza software A dell'applicazione critica e iniettare direttamente i guasti MG a livello microscopico in locazioni LC, e misurare un corrispondente effetto finale FFM in tale istanza software A di detta applicazione, il procedimento, con riferimento anche a quanto illustrato nel diagramma di flusso di Figura 4, prevede di effettuare l'iniezione di guasti, selezionando un determinato guasto microscopico MAXP da iniettare, di selezionare in una libreria di mutanti software un mutante software corrispondente a detto guasto microscopico MAXP da iniettare, applicare il mutante selezionato IMUT a detta istanza software A per ottenere un istanza mutata AM, che è quindi una funzione AM(MAXP) di tale guasto microscopico MAXP da iniettare. Si procede quindi a simulare detto sistema o circuito elettronico eseguente detta istanza mutata, in particolare in un determinato scenario di test, ossia impostando determinati parametri della simulazione e

ponendo il modello software che simula il sistema o circuito elettronico in una certa condizione iniziale. Preferibilmente lo scenario è impostato in funzione del guasto microscopico MAXP da iniettare, e misurare l'effetto finale corrispondente all'istanza mutata.

In altre parole, l'operazione di iniettare guasti MG di figura 2, viene realizzata attraverso l'iniezione di un'istanza software dell'applicazione mutata AM corrispondente al guasto da iniettare MAXP.

L'applicazione mutata AM genera una reazione del sistema che viene valutata a uno o più punti di osservazione O e punti di decisione D per valutare modi fallimento radice RFM e, infine, i conseguenti effetti finali FFM. Il procedimento descritto viene preferibilmente applicato una pluralità di volte, secondo un numero di scenari, e inoltre può venire applicato per ciascun guasto in una lista di guasti individuati tramite analisi.

La Figura 4 illustra uno schema a blocchi di un procedimento per misurare l'effetto di guasti hardware microscopici in applicazioni ad elevata complessità implementate in un sistema elettronico hardware, indicato complessivamente con il riferimento numerico 200.

Tale sistema elettronico può corrispondere, come detto, ad esempio sistemi elettronici per applicazioni automobilistiche di assistenza al conducente e guida parzialmente o completamente automatica, che comprende un architettura di elaborazione che opera in sicurezza funzionale.

Con 210 è indicata un passo di analisi dettagliata dello hardware utilizzato dal sistema elettronico 11 sotto esame. Tale analisi può prevedere di operare in particolare essere attuata come descritto nel brevetto US7937679, in

particolare nella descrizione pertinente alla figura 3 di tale documento per quanto riguarda la suddivisione del sistema 11 in parti elementari o zone sensibili. A ciascuna zona sensibile corrisponde un record di database con valori di safe e dangerous failure rate, divisi fra guasti transienti e permanenti.

Tale passo di analisi 210 comprende di costruire una struttura dati, ad esempio in forma di tabella TMG, mostrata in Figura 5, che per ogni riga della tabella (record della struttura dati) corrispondente a ciascun possibile guasto hardware microscopico MG, in particolare a ciascun guasto hardware microscopico MG che si considera per la possibile iniezione in una simulazione, comprende una pluralità di colonne (campi) comprendenti:

- la descrizione o nome del guasto hardware microscopico MG;
- la sua tipologia TyMG (se permanente o transiente)
- il modo di fallimento FMMG
- la probabilità di guasto relativa, PMGr, ossia la probabilità di guasto del dato guasto MG rispetto alla probabilità di guasto totale del sistema 11.

Ad esempio, come mostrato nella Figura 5, dov'è mostrata la tabella TMG ed è dettagliato un record di tale tabella relativo a un dato guasto microscopico MG, l'analisi identifica nel record di un guasto hardware MG "Stuck-at registro dell'indirizzo di salto di programma", di tipo TyMG permanente, con modo di fallimento FMMG "Programma salta sempre alla solita posizione" il 25% della probabilità relativa PMGr di guasto del sistema elettronico. In altre parole, la probabilità di guasto è causata da uno "stuck-at" del registro dell' indirizzo di

salto di programma di un processore, che a sua volta a livello funzionale può causare che il programma salti sempre alla solita posizione.

Dunque il passo 210 ha come uscita una struttura dati dei guasti microscopici TMG comprendente record di campi con informazioni per ciascun guasto hardware microscopico MG che viene identificato o si considera potenzialmente iniettabile.

In parallelo, come mostrato in figura 1, in un passo 230 viene costruita una libreria LM di mutanti software, ossia una struttura di dati, mostrata anch'essa in forma tabellare in Figura 5, dove ogni riga della tabella (record della struttura dati) corrisponde a un possibile mutante software MUT, definito attraverso una pluralità di colonne (campi) comprendenti:

il tipo di mutante software TMUT;

la persistenza PMUT (persistente o temporaneo);

il grado di applicazione QMUT del mutante, ossia in qual quantità va applicato il mutante MUT.

La tabella LM comprende anche una colonna MUT con una descrizione/identificativo del mutante, in questo caso un semplice numero d'ordine.

La mutazione software, come accennato, è in generale una tecnica nota. Un mutante software MUT è una mutazione locale di un programma.

Ad esempio il programma:

```
if (a && b) {
    c = 1;
} else {
    c = 0;
}
```

viene mutato in:

```
if (a || b) {
    c = 1;
} else {
    c = 0;
}
```

tramite una mutazione di operazione da && a ||.

Come indicato il tipo di mutante software TMUT può assumere molteplici valori, in particolare tre, mutazione di valore, mutazione di decisione e mutazione di operazione. Il campo persistenza PMUT può essere persistente o temporaneo. Il campo grado di applicazione QMUT può avere differenti valori a seconda di quanto viene applicato il mutante MUT, ad esempio in base al numero di punti di decisione, da 'tutti i punti di decisione', ossia da tutti i punti di salto di programma a numeri inferiori di punti di salto di programma, o decisione, secondo il numero che si desidera impostare.

La costruzione della libreria LM infatti può avvenire tramite la descrizione di mutanti ricavati dall'esperienza diretta dell'utilizzatore del procedimento di misura rispetto all'applicazione A sotto esame oppure tramite l'utilizzo di mutanti già descritti in letteratura come ad esempio quelli indicati in testi come Yue Jia, Mark Harman (September 2009). "*An Analysis and Survey of the Development of Mutation Testing" (PDF)*. CREST Centre, King's College London, Technical Report TR-09-06, disponibile all'URL http://crest.cs.ucl.ac.uk/fileadmin/crest/sebasepaper/JiaH10.pdf

L'organizzazione della libreria LM come mostrato in Figura 5, definendo ciascun mutante software MUT impiegabile secondo tipologia di mutante TMUT, persistenza

del mutante PMUT e quantità di mutanti da iniettare QMUT permette un collegamento il più possibile automatico tra mutanti MUT e guasti hardware microscopici MG.

E' previsto quindi di ordinare in un passo 220 tutti i guasti hardware microscopici MG identificati al passo 210, ossia nella struttura TMG precedentemente discussa, in funzione della loro probabilità di guasto relativa e eseguire un processo di mutazione 260 che comprende un passo 261 in cui, a partire dal guasto microscopico più probabile MAXP, si ricerca, nella libreria di mutanti LM, il mutante MUT che vi corrisponde in base a:

- criterio di persistenza: associa il campo tipologia guasto TyMG e il campo persistenza PMUT, ad esempio un guasto hardware con tipologia TyMG permanente deve corrispondere ad un mutante con persistenza PMUT persistente. Esempi di altri valori che può assumere il campo PMUT sono persistenza transitoria e persistenza intermittente;
- criterio di modo di fallimento: associa il campo modo di fallimento FMMG e il campo tipo di mutante TMUT, ad esempio un guasto che comporta un modo di fallimento FMMG per cui il programma salti sempre alla solita posizione deve corrispondere ad un tipo di mutante TMUT quale un mutante di decisione. Esempi di altri valori che può assumere il campo TMUT sono mutante di operazione aritmetica (ad es. scambio di somma con prodotto), mutante di relazione (ad es. scambio di maggiore con minore), mutante di scopo (ad es. scambio di una variabile con un'altra), mutante di cancellazione (ad es. cancellazione di un passo del programma), mutante di inserimento (ad es. inserimento o duplicazione di un passo di programma);

• criterio di quantità: associa il campo probabilità di guasto relativa PMGr e il campo quantità di mutanti da iniettare QMUT, ad esempio un guasto microscopico molto probabile, cioè con probabilità di guasto relativa PMGr molto alta, ad esempio sopra una data soglia, deve corrispondere ad un mutante che opera su tutti i punti di salto del programma, ossia con campo QMUT così impostato. Il valore associato al campo quantità di mutanti da iniettare QMUT può essere relativo ad una distribuzione uniforme totalmente casuale dei mutanti (ad esempio rispetto ad un tipo di mutante TMUT di operazione aritmetica, vengono mutate il 25% di tutte le operazioni aritmetiche individuate nel programma) oppure ad una distribuzione uniforme, ma egualmente distribuita su vari insiemi di passi di programma (ad esempio rispetto ad un tipo di mutante TMUT di operazione aritmetica, vengono mutate il 25% delle operazioni aritmetiche individuate nel programma ma tale 25% viene distribuito su cinque insiemi corrispondenti ad esempio a cinque sotto programmi del programma principale) oppure ancora ad una distribuzione non uniforme ma concentrata sui punti di programma di maggiore criticità, ossia relativi a passi di programma che si ritengono (per esperienza o in base a precedenti cicli di mutazione) particolarmente rischiosi per la sicurezza del sistema.

Tali criteri sono impostabili ad esempio tramite tabelle di look-up in cui vengono memorizzate le associazioni fra determinati valori (o range di valori) assunti dei campi della tabella guasti microscopici TMG, rispettivamente TyMG, FMMG e PMGr, e i valori dei campi della libreria mutanti, rispettivamente TMUT, PMUT, QMUT. Ad esempio, fra tipo guasto TyMG e tipo mutante TMUT si

potrà avere un'associazione uno-uno, così come fra modo di fallimento FMMG e persistenza mutante PMUT, mentre range di valori di probabilità PMGr corrispondono a diversi valori di quantità di mutanti da iniettare QMUT.

Preferibilmente i criteri vengono verificati in maniera sequenziale, ossia in base al campo tipologia guasto TYMG viene individuato nella libreria LM un gruppo di record (in figura 5 i record 2,3,4), aventi un determinato tipo di mutante TMUT. Quindi fra tali record si seleziona il record con persistenza PMUT corrispondente al modo di fallimento del guasto MG selezionato, ossia il guasto microscopico più probabile MAXP (record 2 e 3). Infine viene selezionato fra tali record il record avente quantità di mutanti da iniettare QMUT corrispondente alla probabilità relativa PMGr del guasto microscopico più probabile MAXP.

Dunque, quanto finora discusso con riferimento a figura 5 mostrato come dal guasto MG indicato, ossia il guasto microscopico più probabile MAXP, viene selezionato un record di un mutante MUT con determinati valori corrispondenti nei campi della lista LM. In base alla selezione, viene prelevato dalla libreria LM un mutante selezionato IMUT corrispondente ossia a uno script software che è configurato per applicare, in un passo 262 la mutazione selezionata all'applicazione A, determinando un applicazione mutata AM. Ad esempio, data un'istanza (programma software) di applicazione A, come quella qui sotto rappresentata in codice C++:

```
#include <stdio.h>

#include "new.h"
#include "Object.h"
#include "Set.h"

int main ()
{   void * s = new(Set);
    void * a = add(s, new(Object));
    void * b = add(s, new(Object));
    void * c = new(Object);

    if (contains(s, a) && contains(s, b))
        puts("ok");

    if (contains(s, c))
        puts("contains?");

    if (differ(a, add(s, a)))
        puts("differ?");

    if (contains(s, drop(s, a)))
        puts("drop?");

    delete(drop(s, b));
    delete(drop(s, c));

    return 0;
}
```

la mutazione iniettata IMUT, mutazione di decisione determina nelle decisioni, ossia i costrutti if dell'istanza A il seguente risultato,

```
if (contains(s, a) && contains(s, b));
    puts(*contains?*);
if (contains(s, c));
    puts(*contains?*);
if (differ (a, add(s, a)));
    puts(*contains?*);
if (contains(s, drop(s, a)));
```

ossia un'istanza di applicazione mutata AM.

Una volta identificato il mutante IMUT da iniettare nell'istanza software dell'applicazione A per ottenere l'istanza mutata AM, viene effettuata in un passo 270 la

misura tramite simulazione dell'istanza di applicazione mutata AM nel sistema 11, come mostrato in figura 4. Come detto infatti, la procedura di misura comprende l'esecuzione di un modello software (programma) dell'applicazione, in questo caso applicazione mutata AM, eseguita dal sistema elettronico 11 sotto esame e la misura dell'effetto finale FFM di tale simulazione.

Come mostrato in figura 4, il passo 270 di misura prevede di eseguire in precedenza di due ulteriori passi:

- un passo 240 di preparazione del modello software dell'applicazione, qui identificato con A . Questo è un passo ben noto all'esperto del ramo: ad esempio può essere un modello in C++ eseguito su processori INTEL. Tale passo può essere eseguito una volta, prima dell'esecuzione del procedimento di misura qui descritto.
- un passo 250 di selezione/ creazione di scenari di test ST con cui sollecitare il modello software A dell'applicazione. Per scenari di test ST si intende in generale il portare il detto modello software A dell'applicazione in una determinata condizione iniziale, ad esempio nel caso di un veicolo a guida autonoma in una condizione corrispondente ad una strada con la presenza di un ostacolo ma distanza superiore alla soglia di decisione per la frenata automatica e l'impostare condizioni di simulazione determinati da valori di parametri impostabili nella simulazione, ad esempio nel caso sopra descritto impostando una velocità del veicolo superiore ai 50 Km/h.

Tale passo 250 è eseguito in funzione della tipologia del guasto microscopico MAXP da iniettare tramite l'applicazione mutata AM. Ad esempio, nel caso del guasto microscopico MAXP mostrato in Figura 5, ossia lo "stuck-at" del registro dell' indirizzo di salto di programma di un

processore, viene identificata la funzione che tale processore compie nell'ambito dell'applicazione e conseguentemente per negazione viene creato uno scenario di test.

Una volta che si abbia il modello software dell'applicazione dal passo 240 e l'insieme degli scenari di test dal passo 250, si può procedere alla misura 270 che comprende, a seguito della mutazione 262 il modello o istanza software A tramite il mutante selezionato IMUT, ottenendo il modello mutato AM, di simulare il funzionamento del sistema 11 che implementa il modello software mutato AM di applicazione.

Il passo 270, per una singola simulazione in un dato scenario di test ST, ha come uscita un effetto finale FFM.

E' quindi previsto un passo 280 di classificazione di risultati R, ossia gli effetti finali FFM rilevati, della simulazione 270 secondo classi di effetto Ca, Cs, Cw in risultati classificati RC.

E' quindi previsto di eseguire un passo 290 per calcolare indici di sicurezza del guasto microscopico MAXP in base al quale è stato selezionato il mutante iniettato IMUT, che in funzione dei risultati classificati RC ottenuti per detto mutante iniettato IMUT. Il passo 290 in questo modo, associando valori di indici di sicurezza al guasto microscopico MAXP, a cui è già associata una probabilità di guasto relativa PMGr, permette successivamente di impiegare in modo collegato rispetto allo stesso guasto microscopico, MAXP, tali indici di sicurezza, ottenuti dai risultati classificati RC secondo le classi Ca, Cs, Cw di effetto finale FFM, e probabilità di guasto relativa PMGr per calcolare grandezze rappresentative dell'effetto di tale guasto microscopico

MAXP iniettato. Specificamente, gli indici di sicurezza calcolati al passo 290 sono la frazione di guasti sicuri S e la copertura diagnostica DC. Preferibilmente, a seguito del passo 290, può venire calcolata, in un passo 295, per il guasto microscopico MAXP, una corrispondente grandezza indicativa dell'effetto quale, ad esempio, la probabilità di guasto non rilevato, calcolata in particolare come prodotto della probabilità di guasto relativa PMGr, del complemento della frazione di guasti sicuri (1-S) e del complemento della copertura diagnostica, ossia PMGr *(1-S) * (1-DC).

Preferibilmente il passo 290 viene eseguito al termine del numero NST di iterazioni del passo di misura 270 derivanti dalla sequenza di test impostata al passo 250.

Al fine di spiegare la procedura, a puro titolo di esempio, si consideri l'applicazione A schematizzata in Figura 6, composta da alcune funzioni software Fi come:

- una funzione F1 che compie la lettura dell'immagine catturata da una video camera CMOS montata sul veicolo;
- una funzione F2 di elaborazione dell'immagine per l'identificazione di un eventuale ostacolo;
- una funzione F3 che effettua il calcolo della distanza dell'eventuale ostacolo rispetto al veicolo;
- una funzione F4 che nel caso di mancanza di ostacoli, mantiene la velocità del veicolo;
- una funzione F5 che nel caso di una presenza di ostacolo a distanza ravvicinata effettua una frenata di emergenza;
- una funzione F6 che in parallelo alle altre compie una supervisione, ossia nel caso in cui siano

rilevati degli errori nel funzionamento del sistema, effettua una procedura di emergenza come ad esempio la riduzione della velocità del veicolo ed il suo parcheggio al bordo della strada.

- Nell'ambito di questo esempio, supponendo che il processore soggetto al guasto da iniettare MAXP stuck-at sia quello che esegue il calcolo della distanza di un eventuale ostacolo, uno scenario di test ST è una condizione in cui vi sia la presenza di un ostacolo lontano, in quanto quello che si vuole verificare è se un eventuale guasto hardware può causare che il sistema erroneamente scambi un ostacolo lontano per un ostacolo vicino, e quindi inizi una frenata di emergenza quando non necessario.

La Figura 7 esemplifica il passo 280 di categorizzazione, con riferimento all'applicazione A descritta in Figura 6. L'esecuzione del modello di software mutato AM può causare tre effetti finali FMMb, FMMa, FMMc diversi, rispettivamente effetto di evento atteso, effetto di evento non atteso o errato, effetto di sicurezza, ciascuna corrispondente a una categoria Ca, Cs, Cw di effetto:

- categoria Ca di effetto atteso: la mutazione viene tollerata oppure è senza effetto. Nell'esempio di Figura 7, nonostante la funzione F3 sia stata mutata, l'ostacolo viene sempre ritenuto lontano (decisione D2) e quindi l'effetto finale FMMb è quello atteso, ossia la velocità del veicolo non viene modificata;

- categoria Cw di effetto errato: la mutazione causa un malfunzionamento. Nell'esempio, a causa della mutazione della funzione F3, l'ostacolo lontano viene invece ritenuto

vicino (decisione D1) e quindi l'effetto finale FFMa è diverso da quello atteso, ossia viene effettuata una frenata di emergenza quando non necessario, con conseguente potenziale pericolo per il guidatore che viene disorientato dall'evento non atteso;

- categoria Cs di effetto di sicurezza: la mutazione potrebbe causare un malfunzionamento ma viene rilevata e quindi vengono attuate delle procedure speciali. Nell'esempio, la funzione di supervisione F6 rileva il potenziale malfunzionamento e viene iniziata la procedura di emergenza (denominata evento, o effetto, di sicurezza), ossia il veicolo riduce velocità e parcheggia, informando il guidatore del malfunzionamento del sistema elettronico 11.

Tale passo di categorizzazione 280 viene ripetuto per tutti gli scenari di test ST e per tutti i mutanti selezionati come da iniettare.

La Figura 8 illustra il passo 290, ossia il collegamento della classe Ca, Cs, Cw di effetto finale di un mutante iniettato IMUT con la probabilità di guasto microscopico PMGr del corrispondente guasto da iniettare MAXP.

Nell'esempio cui si riferisce la Figura 8 sono stati effettuati 100 test con una mutazione di decisione persistente in tutti i punti di decisione, come da esempio in Figura 5. Si considera che per 90 volte il risultato di un test è che si sia verificato l'effetto atteso FMMb, per 9 volte il risultato R di un test è che sia verificato l'effetto di sicurezza FMMc, mentre per una volta si è verificato l'evento errato FMMa. I risultati R per ciascun mutante iniettato IMUT sono riportati nella struttura dati di tabella risultati RT, ossia una struttura dati di

classificazione, avete per righe per ciascun mutante iniettato IMUT e per colonne, oltre al nome, descrizione o identificativo del mutante iniettato IMUT, il numero di scenari NST iniettati, ossia il numero di test, per quel mutante IMUT e il risultato complessivo RC, raggruppato in valori secondo le classi Cw, Ca, Cs, ossia nell'esempio Ca=90, Cs=9, Cw=1. Sostanzialemente il risultato complessivo RC è una terna di valori corrispondenti a ciascuna classe (90, 9, 1).

L'operazione di collegamento 290 viene stabilita in questo modo per ciascun guasto da iniettare IMUT:

- la frazione di guasti sicuri (denominata anche fattore S) è calcolata pari al rapporto del numero di test NST effettuati per la mutazione IMUT corrispondente a tale guasto microscopico MG che hanno dato come risultato l'evento atteso FMMb rispetto al numero totale di test per la mutazione IMUT corrispondente a tale guasto microscopico da iniettare MAXP;
- la copertura diagnostica (denominata anche fattore DC) è pari al rapporto del numero di test NST effettuati per la mutazione corrispondente IMUT a tale guasto microscopico MG che hanno dato come risultato l'evento di sicurezza FMMc rispetto al numero di test NST per la mutazione IMUT corrispondente a tale guasto microscopico MG che hanno dato risultato l'evento di sicurezza FMMc o l'evento errato FMMa.

Questi due fattori di sicurezza S e copertura diagnostica DC sono valori determinanti per il calcolo delle metriche di sicurezza funzionale previsti nelle norme citate e in figura 8 è mostrato come l'operazione di collegamento 290 generi una struttura dati collegata ai risultati TMGc, che corrisponde alla tabella TMG cui sono

aggiunte due colonne o campi per i valori di fattore di sicurezza S e copertura diagnostica DC corrispondenti a un dato guasto microscopico MG.

I passi del procedimento 220, 261, 262, 250, 270, 280, 290 possono venire ripetuti per tutti i guasti microscopici nella tabella guasti microscopici TMG con probabilità relativa PMGr significativa (la simulazione 270 essendo ogni volta applicata su un rispettivo numero NST di scenari di test), ad esempio maggiore di una soglia, ad esempio maggiore di 1%, scegliendo a ogni iterazione come guasto da iniettare MAXP, il guasto MG secondo un ordine di probabilità relativa PMGr decrescente. I passi 230 di preparazione della libreria LM e 240 di preparazione del modello software dell'applicazione A vengono eseguiti preferibilmente una volta sola, precedentemente al passo 220 di identificazione dei guasti più probabili.

Dunque, dalla descrizione risultano chiari i vantaggi dell'invenzione.

La soluzione descritta, in particolare il procedimento e il corrispondente sistema di simulazione che implementa il procedimento, rendono quindi possibile in modo estremamente semplice ed automatico il collegamento tra guasti hardware microscopici e i loro effetti finali. L'utilizzo di mutazioni software permette di eseguire l'iniezione guasti su un modello software che è, anche per applicazioni estremamente complesse, sia estremamente veloce sia capace di modellare, tramite le mutazioni, gli effetti causati da tali guasti hardware microscopici.

La soluzione descritta, oltre alla drastica riduzione dei tempi di misura, permette di identificare in modo chiaro e completo gli effetti finali dei guasti hardware microscopici. Questo permette l'ottimizzazione della

funzione di supervisione, ossia una sua modifica al fine di ottenere – con il minor costo possibile – il valore più alto possibile di copertura diagnostica (DC). In questo modo si ottiene un significativo aumento della sicurezza funzionale del dispositivo e di conseguenza la riduzione degli eventi pericolosi per la vita umana.

La soluzione descritta si applica in modo del tutto analogo ad applicazioni in campo industriale, medicale ed aereonautico.

Naturalmente, fermo restando il principio dell'invenzione, i dettagli e le forme di attuazione possono variare, anche in modo rilevante, rispetto a quanto qui descritto a puro titolo di esempio, senza discostarsi dall'ambito di protezione. Tale ambito di protezione è definito dalle rivendicazioni annesse.

Le strutture dati descritte, in particolare tabelle, possono essere implementate attraverso uno o più database, ospitati su uno o più elaboratori, che possono comprendere l’elaboratore o gli elaboratori che operano da sistema di simulazione.

RIVENDICAZIONI

1. Procedimento per misurare l'effetto di guasti hardware microscopici in applicazioni ad elevata complessità implementate in un sistema elettronico hardware, comprendente operare su un sistema di elaborazione (600) una fase di simulazione (270) di detto sistema o circuito elettronico (11) eseguente una istanza software (A) di detta applicazione, detta fase di simulazione (270) comprendendo iniettare guasti (MG) microscopici e misurare un corrispondente effetto finale (FFM) in detta istanza software (A) di detta applicazione,

caratterizzato dal fatto

detta operazione di iniettare guasti (MG) comprende

selezionare (210, 220) un guasto microscopico (MAXP) da iniettare

selezionare (261) in una libreria di mutanti software (LM) un mutante (IMUT) corrispondente a detto guasto microscopico (MAXP) da iniettare

applicare (262) detto mutante selezionato (IMUT) a detta istanza software (A) per ottenere un istanza mutata (AM)

simulare (270) detto sistema o circuito elettronico (11) eseguente detta istanza mutata (AM) in un determinato scenario di test (ST)

misurare (280, 290) l'effetto corrispondente (FMMa, FMMb, FMMc) all'istanza mutata (AM).

2. Procedimento secondo la rivendicazione 1, caratterizzato dal fatto che comprende

selezionare o creare (232), in funzione del guasto microscopico (MAXP) da iniettare, una sequenza (NST) di scenari (ST) comprendenti parametri di sollecitazione dell'istanza (AM) dell'applicazione critica;

dette operazioni di simulare (270) e misurare (280, 290) essendo eseguita una pluralità di volte secondo detta sequenza (NST) di scenari di test (ST).

3. Procedimento secondo la rivendicazione 1 o 2, caratterizzato dal fatto che

detta operazione di selezionare (210, 220) un guasto microscopico (MAXP) da iniettare comprende

eseguire (210) un'analisi del sistema elettronico (11) per identificare un insieme di guasti hardware microscopici (MG) da iniettare

individuare (220) in detto insieme di guasti hardware microscopici (MG) guasti hardware microscopici più probabili (MAXP);e

detta operazione di selezionare (261) in una libreria di mutanti software (LM) un mutante (IMUT) corrispondente a detto guasto microscopico (MAXP) da iniettare comprende

costruire una libreria (LM) di mutanti software, in particolare organizzata per tipologia (TMUT), persistenza (PMUT) e quantità (QMUT) di mutanti;

selezionare da detta libreria (LM) dei mutanti (IMUT) corrispondenti ai potenziali guasti hardware microscopici più probabili (MAXP).

4. Procedimento secondo la rivendicazione 1, caratterizzato dal fatto che detta operazione di misurare (280, 290) l'effetto corrispondente (FMMa, FMMb, FMMc) all'istanza mutata (AM)comprende rilevare detto effetto finale (FMMa, FMMb, FMMc) e classificarlo (280) secondo classi di effetto finale (Ca, Cs, Cw), in particolare comprendenti classi di evento atteso (Ca), evento di sicurezza (Cs) ed evento errato (Cw).

5. Procedimento secondo la rivendicazione 4, caratterizzato dal fatto che comprende un passo di calcolo

(290) di indici di sicurezza (S, DC) del guasto microscopico (MAXP) in funzione di risultati (RC) di detto passo di classificazione (280).

6. Procedimento secondo la rivendicazione 5, caratterizzato dal fatto che comprende inoltre calcolare (295) grandezze rappresentative dell'effetto del guasto microscopico (MAXP) da iniettare in funzione di detti indici di sicurezza (S, DC).

7. Procedimento secondo la rivendicazione 5 o 6, caratterizzato dal fatto che detto passo di calcolo (290) di indici di sicurezza (S, DC) comprende

calcolare il numero di eventi attesi (Ca), eventi di sicurezza (Cs) ed eventi errati (Cw) per una sequenza di test (NST) effettuata con un determinato mutante (IMUT);

calcolare in funzione di detti numero di eventi attesi (Ca), eventi di sicurezza (Cs) ed eventi errati (Cw) valori di frazione di guasti sicuri (S) e copertura diagnostica (DC) e associare detti indici (S, DC) a detto guasto microscopico (MAXP) selezionato.

8. Procedimento secondo una delle rivendicazioni precedenti da 2 a 7, caratterizzato dal fatto che detta operazione di individuare (220) in detto insieme di guasti hardware microscopici (MG) guasti hardware microscopici più probabili (MAXP) comprende ordinare detti guasti hardware microscopici (MG) per valore di probabilità di guasto relativa (PMGr) e eseguire almeno dette operazioni di selezionare (261) in una libreria di mutanti software (LM) un mutante (IMUT) corrispondente a detto guasto microscopico (MAXP) da iniettare, applicare (262) detto mutante selezionato (IMUT) a detta istanza software (A) per ottenere un istanza mutata (AM), simulare (270) detto sistema o circuito elettronico (11) eseguente detta istanza

mutata (AM) in un determinato scenario di test (ST), misurare (280, 290) l'effetto corrispondente (FMMa, FMMb, FMMc) all'istanza mutata (AM), per ciascuno di detti guasti hardware microscopici aventi valore di probabilità di guasto relativa (PMGr) maggiore di una soglia determinata.

9. Procedimento secondo una delle rivendicazioni precedenti, caratterizzato dal fatto che detta operazione di analizzare (210) comprende costruire una struttura dati dei guasti microscopici (TMG), comprendente record per ciascun possibile guasto hardware microscopico (MG) avente campi indicanti:

il dato guasto hardware microscopico (MG)

la sua tipologia (TyMG)

il modo di fallimento (FMMG)

la probabilità di guasto relativa (PMGr)

10. Procedimento secondo una delle rivendicazioni precedenti, caratterizzato dal fatto che detta operazione di classificare (280) comprende raccogliere i risultati (R) dell'operazione di simulazione (270) in una struttura dati di classificazione (RT) avente record per ogni mutante iniettato (IMUT) comprendenti campi (NST) relativi al numero di test della sequenza e campi (RC) relativi al numero di effetti in ciascuna determinata categoria (Ca, Cs, Cw).

11. Procedimento secondo la rivendicazione 10, caratterizzato dal fatto che detto passo di calcolo (290) di indici di sicurezza (S, DC) comprende calcolare detti indici di sicurezza (S, DC) per un determinato guasto (MAXP) da iniettare associato a un determinato mutante iniettato (IMUT) in funzione dei valori contenuti in detta struttura dati di classificazione (RT) per detto mutante (IMUT) e di inserire detti valori in una ulteriore

struttura dati (TMG*) ottenuta da detta struttura dati dei guasti microscopici (TMG) aggiungendo per ogni record relativo a un guasto microscopico (MG) campi relativi ai detti indici di sicurezza, in particolare indici di frazione di guasti sicuri (S) e copertura diagnostica (DC).

12. Procedimento secondo la rivendicazione 6 o seguenti, caratterizzato dal fatto che detto passo di calcolare (295) grandezze rappresentative dell'effetto del guasto microscopico (MAXP) da iniettare in funzione di detti indici di sicurezza (S, DC) comprende calcolare una probabilità di guasto non rilevato, come prodotto della probabilità di guasto relativa (PMGr), del complemento della frazione di guasti sicuri (1-S) e del complemento della copertura diagnostica (1-DC).

13. Sistema di simulazione di guasti in sistemi elettronici implementanti applicazioni (A) critiche, comprendente un sistema di elaborazione configurato per operare una fase di simulazione (270) di detto sistema o circuito elettronico (11) eseguente una istanza software (A) di detta applicazione, detta fase di simulazione (270) comprendendo iniettare guasti (MG) a livello microscopico e misurare un corrispondente effetto finale (FFM) in detta istanza software (A) di detta applicazione, caratterizzato dal fatto che il sistema (600) è configurato per eseguire le operazioni del procedimento secondo una o più delle rivendicazioni da 1 a 12.

14. Prodotto informatico caricabile nella memoria di almeno un elaboratore, il prodotto informatico comprendendo porzioni di codice software adattate per eseguire i passi del procedimento secondo una delle rivendicazioni da 1 a 12 nel momento in cui il programma viene eseguito su almeno un computer.

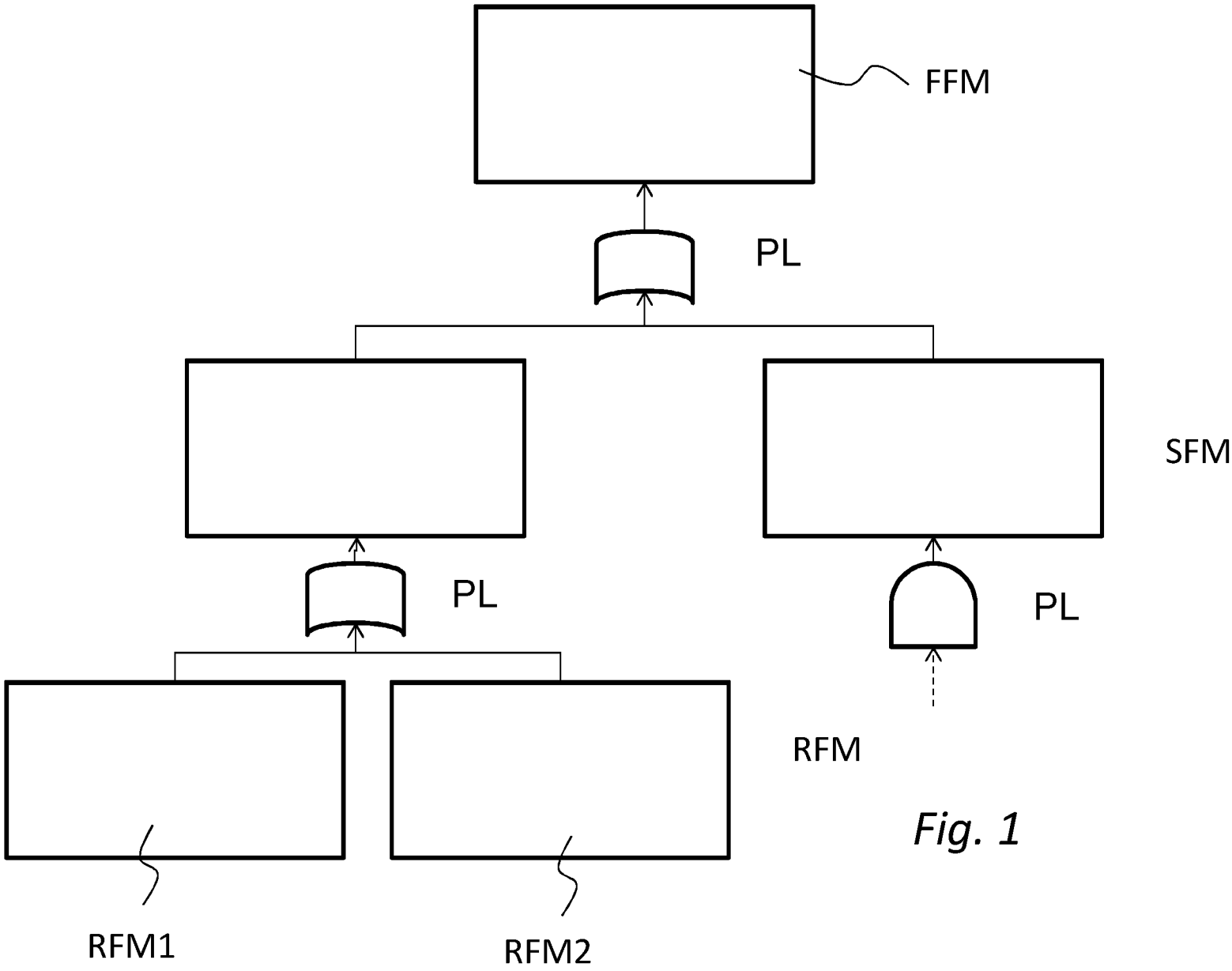


Fig. 1

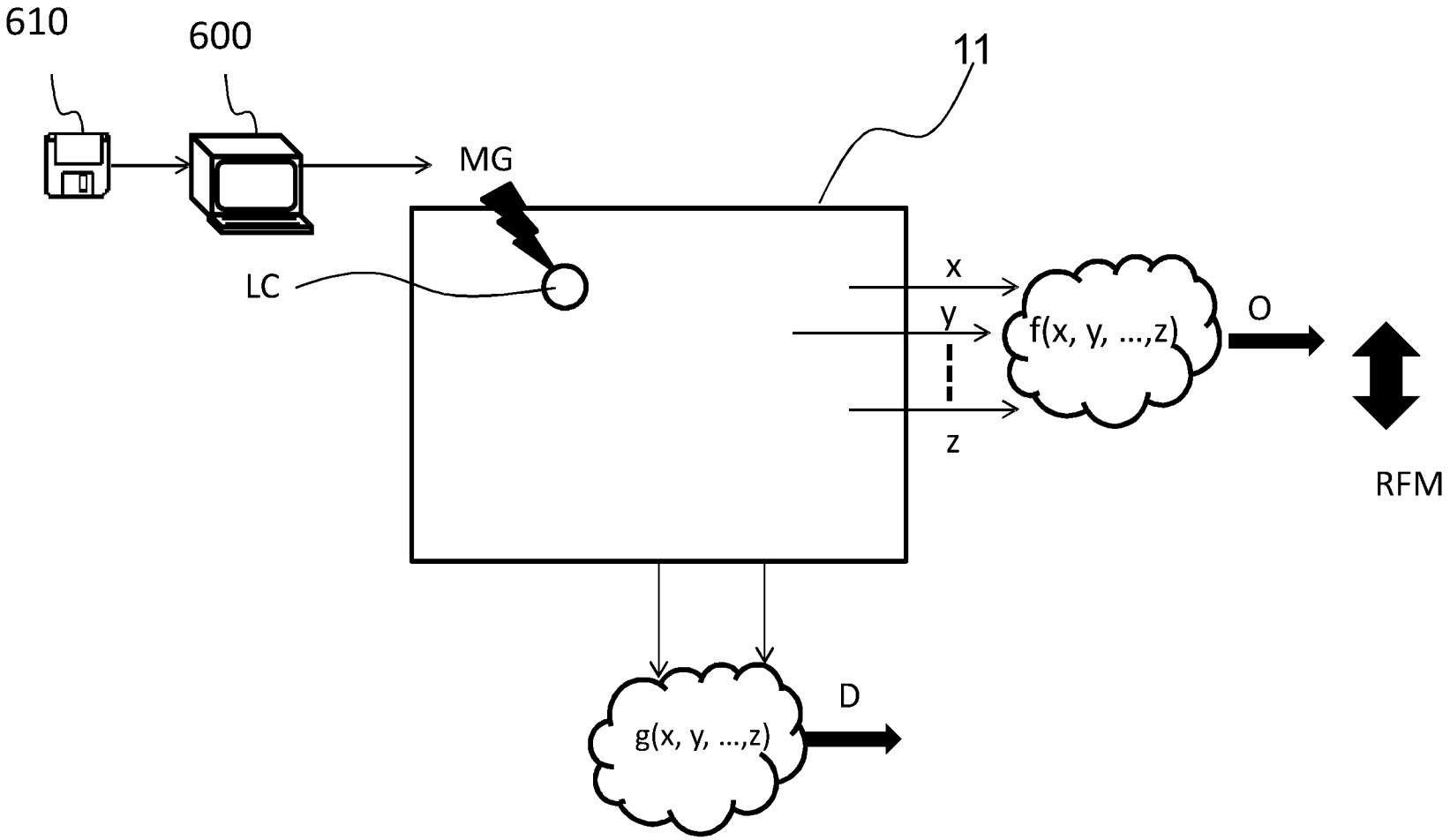


Fig. 2

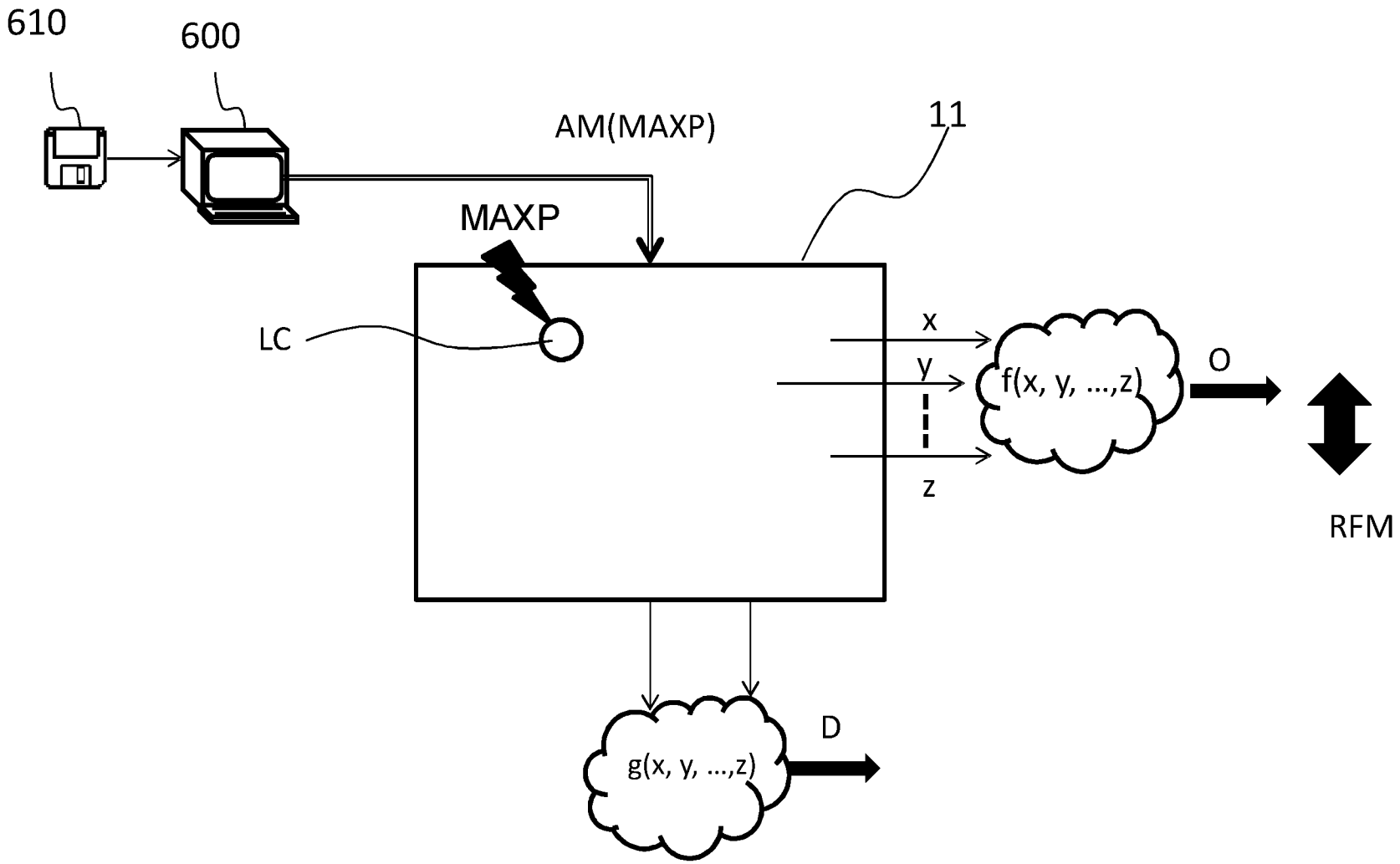


Fig. 3

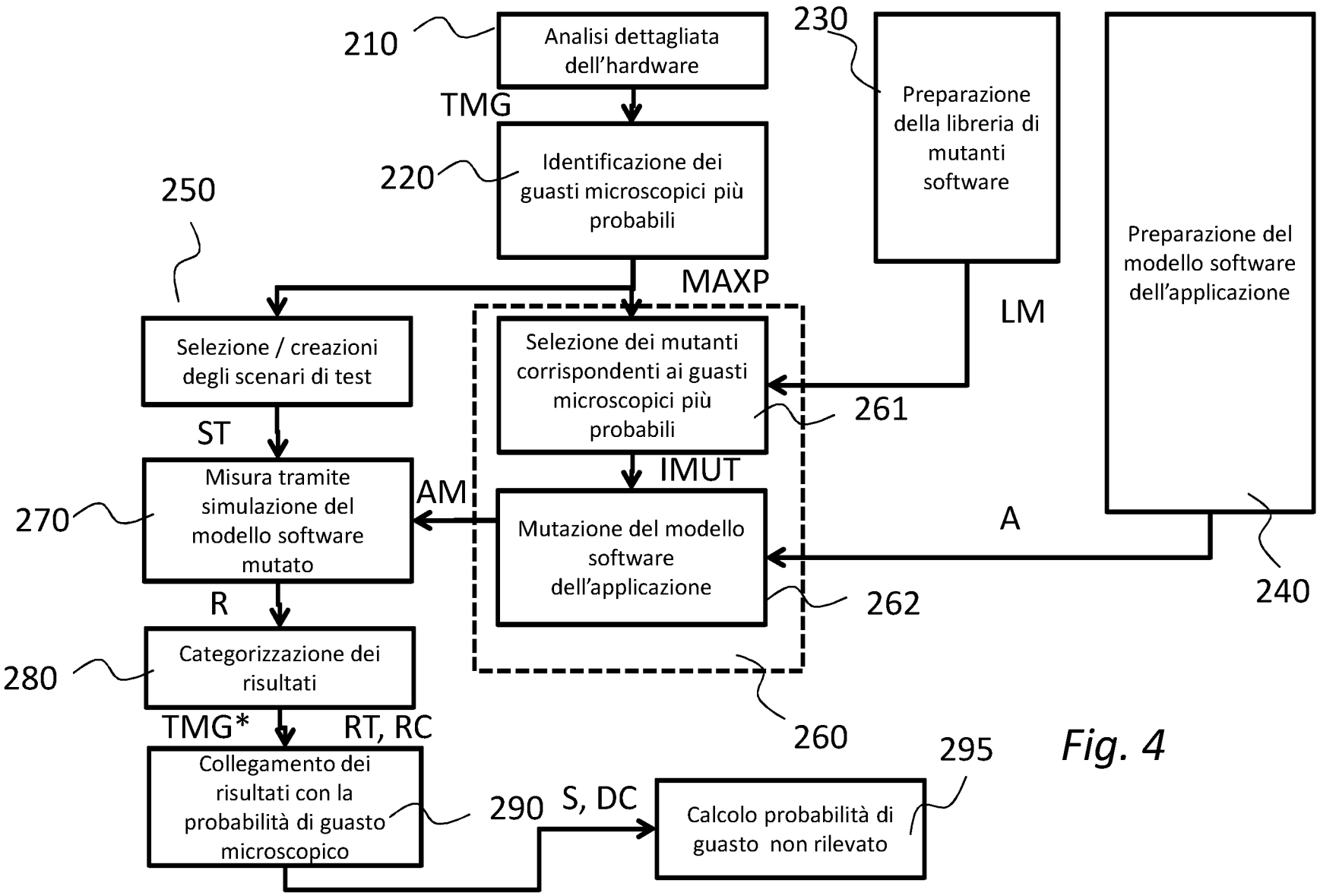

210
Analisi dettagliata dell'hardware
TMG
220
Identificazione dei guasti microscopici più probabili
230
Preparazione della libreria di mutanti software
Preparazione del modello software dell'applicazione
250
MAXP
LM
Selezione / creazioni degli scenari di test
Selezione dei mutanti corrispondenti ai guasti microscopici più probabili
261
ST
IMUT
270
Misura tramite simulazione del modello software mutato
AM
Mutazione del modello software dell'applicazione
A
240
262
R
280
Categorizzazione dei risultati
260
TMG*
RT, RC
Collegamento dei risultati con la probabilità di guasto microscopico
290
S, DC
295
Calcolo probabilità di guasto non rilevato

Fig. 4

Fig. 5

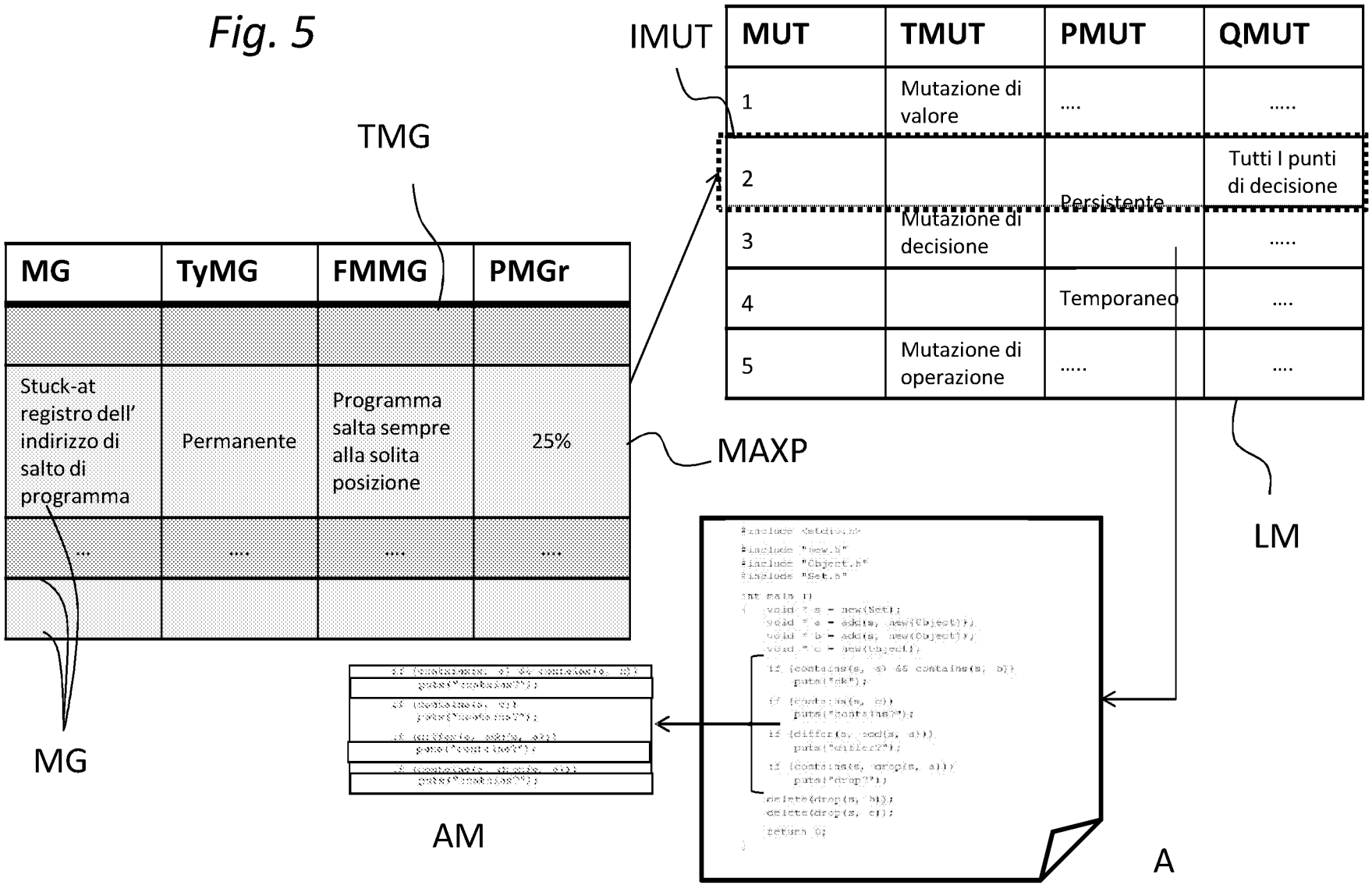
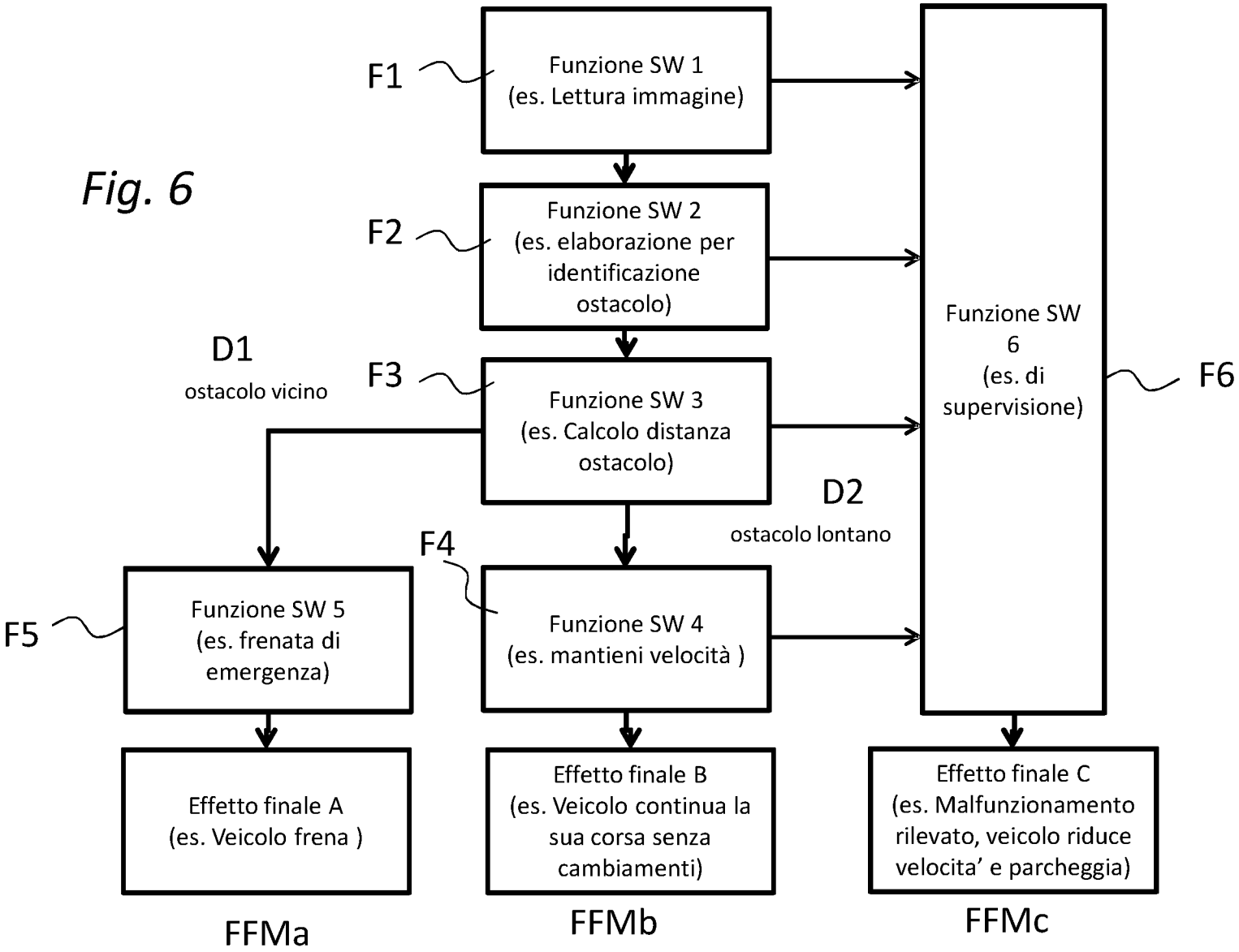

IMUT
TMG
MG
TyMG
FMMG
PMGr
Stuck-at registro dell' indirizzo di salto di programma
Permanente
Programma salta sempre alla solita posizione
25%
MAXP
MUT
TMUT
PMUT
QMUT
1
Mutazione di valore
....
.....
2
Persistente
Tutti I punti di decisione
3
Mutazione di decisione
.....
4
Temporaneo
....
5
Mutazione di operazione
.....
....
LM
AM
A

Fig. 6


F1
Funzione SW 1
(es. Lettura immagine)
F2
Funzione SW 2
(es. elaborazione per identificazione ostacolo)
D1
ostacolo vicino
F3
Funzione SW 3
(es. Calcolo distanza ostacolo)
D2
ostacolo lontano
F4
F5
Funzione SW 5
(es. frenata di emergenza)
Funzione SW 4
(es. mantieni velocità)
Funzione SW 6
(es. di supervisione)
F6
Effetto finale A
(es. Veicolo frena)
Effetto finale B
(es. Veicolo continua la sua corsa senza cambiamenti)
Effetto finale C
(es. Malfunzionamento rilevato, veicolo riduce velocita' e parcheggia)
FFMa
FFMb
FFMc

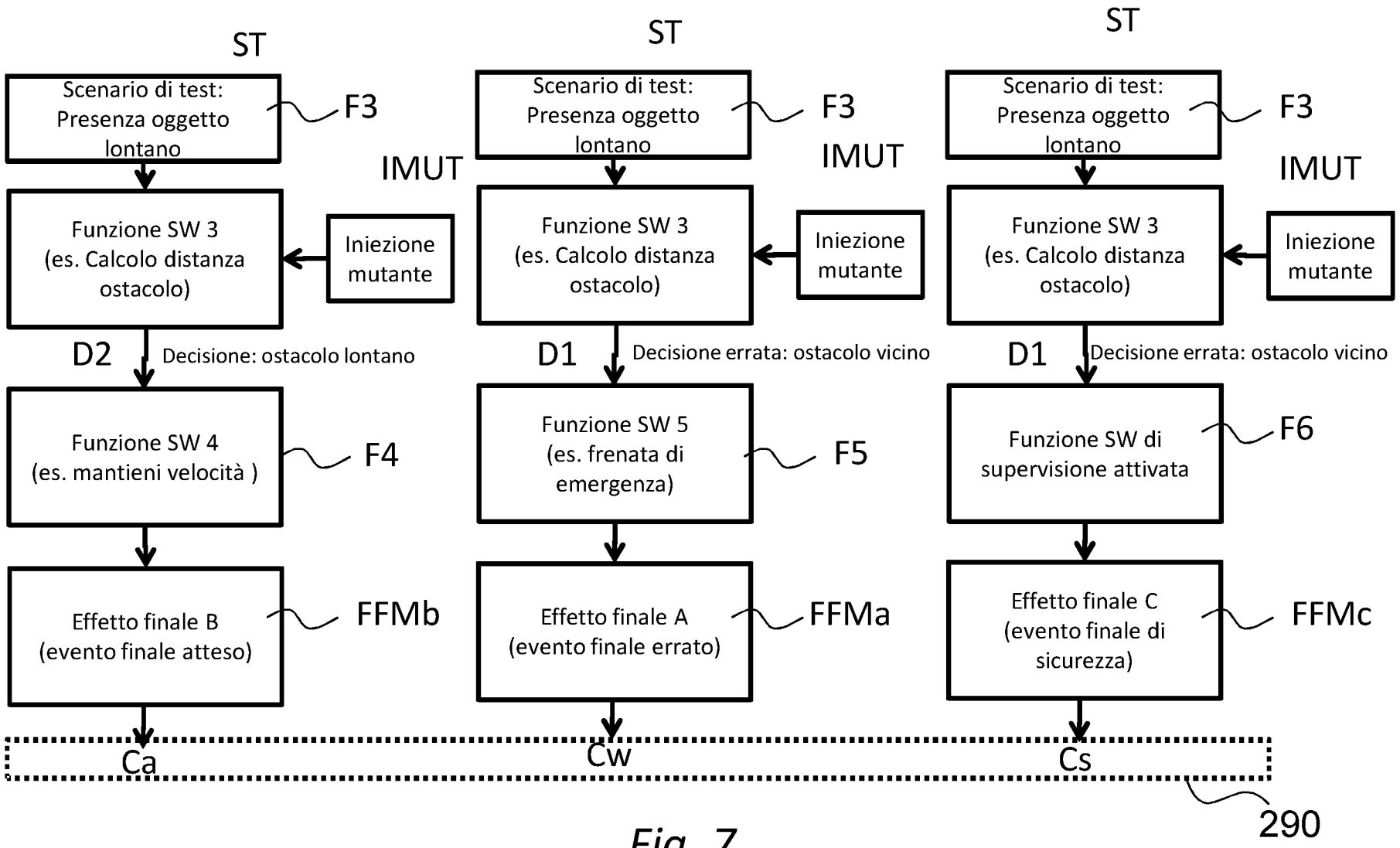

ST
Scenario di test: Presenza oggetto lontano
F3
IMUT
Funzione SW 3 (es. Calcolo distanza ostacolo)
Iniezione mutante
D2
Decisione: ostacolo lontano
Funzione SW 4 (es. mantieni velocità)
F4
Effetto finale B (evento finale atteso)
FFMb
Ca
ST
Scenario di test: Presenza oggetto lontano
F3
IMUT
Funzione SW 3 (es. Calcolo distanza ostacolo)
Iniezione mutante
D1
Decisione errata: ostacolo vicino
Funzione SW 5 (es. frenata di emergenza)
F5
Effetto finale A (evento finale errato)
FFMa
Cw
ST
Scenario di test: Presenza oggetto lontano
F3
IMUT
Funzione SW 3 (es. Calcolo distanza ostacolo)
Iniezione mutante
D1
Decisione errata: ostacolo vicino
Funzione SW di supervisione attivata
F6
Effetto finale C (evento finale di sicurezza)
FFMc
Cs
290

Fig. 7

RT

IMUT	NST	RC (Ca, Cs, Cw)
Mutazione di decisione persistente in tutti I punti di decisione	100	(90, 9, 1) 90 = Ca 9 = Cs 1 = Cw
Mutazione di operazione	.....	....
Mutazione di valore		

Fig. 8