





Estudio de la robustez del « control inteligente » frente a fallos inducidos por las radiaciones

Raoul VELAZCO Directeur de Recherches au CNRS Laboratoire TIMA – CNRS – UGA

Plan

- Introducción
- Efectos de las radiaciones en circuitos y sistemas integrados
- Redes de neuronas artificiales
- Problema estudiado « clasificación de texturas »
- El proyecto MPTB
- El « fuzzy control »
- Hopfield Neural Networks en FPGAs (HNNs)
- Conclusiones y perspectivas

Introducción

Las agencias espaciales se enfrentan a dos problemas :



- Reducciones de presupuestos militares
- Miríadas de nanosatélites de comunicaciones
 Vivimos una era en que los medios de comunicación tienen un crecimiento exponencial
- Los progresos de las tecnologías de fabricación hacen que los circuitos son sensibles a los efectos de las partículas presentes en el entorno natural en que operan

El control inteligente

Las redes de neuronas artificiales y la lógica difusa son potencialmente robustas y aptas a tratar problemas difíciles de resolver por medios clásicos



Sus implantaciones en circuitos dedicados o en procesadores los hace buenos candidatos para ser embarcados en aplicaciones espaciales

Plan

- Introducción
- Efectos de las radiaciones en circuitos y sistemas integrados
- Redes de neuronas artificiales
- Problema estudiado « clasificación de texturas »
- El proyecto MPTB
- El « fuzzy control »
- Hopfield Neural Networks en FPGAs (HNNs)
- Conclusiones y perspectivas

El entorno radiativo espacial



El entorno radiactivo espacial

Entornos radiactivos

Espacio:

Galactic cosmic rays (GCR) Viento Solar (Solar Wind) Actividad Solar Cinturones de radiaciones

Radiaciones en la atmósfera terrestre :

Los rayos cósmicos (helio, protones, electrones & iones pesados) interactúan con las partículas presentes en la atmósfera y provocan una « ducha » permanente (neutrones, protones, etc...) Ejemplo: en Nueva York 14 neutrones/hora x cm2 energía 400 meV









Efectos en circuitos integrados



Efectos transistorios o efectos destructivos

Efectos en circuitos integrados

- Efectos singulares SEE (Single Event Effects)
 - SET (Single Event Transient)
 - SEU (Single Event Upset)
 - MBU (Multiple Bit Upset)
 - SEFI (Single Event Funcional Interrupt)
 - SEL (Single Event Latchup)
- Dosis acumulada TID (Total Intagrated Dose)

SEL (Single Event Latchup)

Corto-circuito GND-VCC provocado por la activación de un tiristor parásito presente en los circuitos CMOS bulk



Destructivo Cortar la alimentación del circuito

Single Event Upset (SEU): modificación del contenido de una celda de memoria



Transitorio, no destructivo

Universidad Complutense de Madrid, 17 Nov. 2015

reescribir el valor del bit

Dosis acumulada

Acumulación de cargas a nivel de los óxidos, debida a los pares electrón/hueco atrapados como consecuencia del impacto de partículas energéticas (protones, iones, neutrones....)

Diversos efectos indirectos pueden aparecer :

modificación de los « gains » de los transistores

• Aparición de corrientes de fuga

Pérdida progresiva de la funcionalidad del circuito, que se vuelve no operacional al cabo de cierto tiempo ^{Universidad Complutense de Madrid, 17 Nov. 2015}

Medios de prevención y de protección

- Blindaje (atenuación de las radiaciones)
- Prevención a nivel de los circuitos :
 - tecnología de fabricación (epitaxial, SOI, AsGa)
 - estructural (tamaño de los transistores, resistencias de contra reacción,
 - diseño (celdas memoria *radhard*,....)
- Protección HW/SW a nivel de sistema :
 - Software (redundancia, ECCs,...)
 - Hardware (arquitecturas redundantes DMR, TMR...)
- Utilización de técnicas de "programación inteligente"
 - ANN, Fuzzy models

Medios de predicción

- Modelos del entorno radiativo (CREME, CHIME, AP8, JPL92)
- Ensayos en aceleradores de partículas (iones pesados, protones, neutrones)
 Fórmulas de aproximación empírica (Petersen)

En todos los casos habrá una diferencia entre la predicción y la realidad



Medios de predicción

Avances en las tecnologías de fabricación de circuitos integrados aumento de la sensibilidad a las partículas energéticas presentes en el entorno natural.

La « calificación » es obligatoria :

-Tests en aceleradores de partículas

- Ensayos con laser o en microhaz

-Tests en el entorno real

LAÍ Globosat 01 | IIE | 2008

Plan

- Introducción
- Efectos de las radiaciones en circuitos y sistemas integrados
- Redes de neuronas artificiales
- Problema estudiado « clasificación de texturas »
- El proyecto MPTB
- El « fuzzy control »
- Hopfield Neural Networks en FPGAs (HNNs)
- Conclusiones y perspectivas

El elemento de base de las ANNs (Artificial Neural Networks): la neurona formal

$$y_k = \Phi(\sum_{j=1}^n w_{kj} \cdot x_j - \theta_k)$$





Neurona formal



Red proactiva de una sola capa « perceptrón »



Red proactiva de una sola capa « perceptrón »



Red proactiva con una capa « oculta »



Red proactiva con una capa « oculta » Universidad Complutense de Madrid, 17 Nov. 2015



Red recurrente

Aprendizaje: supervisado o competitivo

Supervisado o « activo »

Entrada

Salida esperada

Dos ejemplos representativos: « Delta » rule, retropropagación del gradiente

No supervisado o « competitivo »

Entrada I

???

(redes de Kohonen, y de Hopfield)

El aprendizaje es seguido por la fase de generalización (o de « rappel ») sobre las entradas « desconocidas » Universidad Complutense de Madrid, 17 Nov. 2015

Implementaciones de las redes de neuronas

Simulaciones en ordenadores secuenciales:

- Ventajas: diferentes topologías de redes neuronales (Hopfield, MLP, ...) y tipos de aprendizaje pueden ser estudiados
 - Inconvenientes: tiempo de ejecución para redes que pueden poseer centenares o miles de neuronas y miles o millones de sinapsis. Varias horas de cálculo...

Robustez de las redes de neuronas

Las redes neuronales son potencialmente robustas debido a:

La redundancia espacial - replicaciones de neuronas y sinapsis

La naturaleza distribuida de la información
 - contribución directa o indirecta a una ínfima parte del resultado final

La contra-reacción en las redes recurrentes

- redundancia temporal intrínseca

Trabajos de investigación sobre la robustez de ANNs respecto a los SEUs (años 90)

- Los estudios efectuados en TIMA se concentraron en:
- Un perceptrón de 50 neuronas.
- una red de 39 neuronas con 2 capas ocultas
- Estudio de las consecuencia de SEUs en
- Las entradas
- Los pesos/umbrales

robustez de 99% en las entradas y 91% para los pesos

Aplicaciones espaciales de las redes neuronales

- Observación de la Tierra : clasificación de suelos o de nubes, estudio del impacto de aerosoles, ...
- Observación del espacio : « silbidos protónicos », detección de exoplanetas, análisis de « sursauts Gamma », ...

Metodología :
1- Recuperación de datos
2- Pretratamiento
3- Selección de subconjuntos de datos interesantes y análisis por una red neuronal
4- Visualización del experimentador

Aplicaciones espaciales de las ANNs



El proyecto FORTÉ

Micro-satélite lanzado el 29 de Agosto 1997, colaboración Los Alamos National Lab & Sandia National Labs que incluyó :

- Receptores radio
- Un captor óptico CCD

• Un clasificador de eventos basado en un microcontrolador (80C31 radhard) y un DSP (TMS320C30 estándar)

El DSP reconoce fenómenos transitorios (silbidos protónicos) vía una red neuronal que siguió su aprendizaje en la Tierra

Plan

- Introducción
- Efectos de las radiaciones en circuitos y sistemas integrados
- Redes de neuronas artificiales
- Problema estudiado « clasificación de texturas »
- El proyecto MPTB
- El « fuzzy control »
- Hopfield Neural Networks en FPGAs (HNNs)
- Conclusiones y perspectivas











Arquitectura de la red neuronal escogida: MLP (Multi-Layer Perceptron)



Capa de entrada 1 de (24x24)

Retina

Capa 2 1 de (8x8)

Aumento de contraste

Capa 3 3 de (7x7)

Convolución

extracción de

características

Capa 4 x7) 3 de (1x1)

Capa de salida (capa 5) 1 de (4x1)

Clasificador 1 de N

Z. Indus Z. Resi Garriga Mar

Arquitectura de la red neuronal escogida



Capa de entrada 1 de (24x24)

Retina

Capa 2 1 de (8x8)

Aumento de contraste

Capa 3 de (7x7)

Capa 4 $3 de (1x1)^{+}$

Capa de salida (capa 5) 1 de (4x1)

3

Z. Indus

Garriga

r→ Z. Resi

Mar

Convolución extracción de características Clasificador 1 de N
Arquitectura de la red neuronal escogida

Una red ANN constituida de :

- 218 neuronas
- 901 «pesos»

Capa 2 1 de (8x8)

Aumento de contraste

Capa 3 3 de (7x7)

Convolución

extracción de

características

Capa 4 (17) 3 de (1x1)

Capa de salida (capa 5) 1 de (4x1)

Clasificador 1 de N

Z. Indus Z. Resi Garriga Mar

Estudio efectuado

Las respuestas de la función de activación (promedio de todas las neuronas)

Función lineal por pedazos

Función umbral

Estudio efectuado Prestaciones de la red (promedio de 100 ejecuciones) en caso de ruido en las imágenes de entrada



Universidad Complutense de Madrid, 17 Nov. 2015

Estudio efectuado Prestaciones de la red (promedio de 100 ejecuciones) en caso de ruido en la capa de salida



Estudio efectuado

3 métodos (media de 100 ejecuciones) en caso de ruido en los parámetros del ANN



Universidad Complutense de Madrid, 17 Nov. 2015

Estudio efectuado Prestaciones de los 3 métodos (media de 100 ejecuciones) en caso de ruido en los parámetros



Estudio efectuado

En caso ruido en los parámetros

- pesos (métodos neuronales)
- matrices prototipo (método estadístico)

la red ANN estudiada es más robusta

En caso de ruido en las entradas

El método estadístico ofrece ligeramente más prestaciones que las dos redes neuronales estudiadas (que dan resultados casi idénticos)

> Las prestaciones en caso de ruido en las entradas depende del problema estudiado y no del método utilizado para resolverlo Universidad Complutense de Madrid, 17 Nov. 2015

El aprendizaje no es perfecto debido a las razones siguientes:

- Deplazamiento por incremento ±q en un espacio discreto del valor de los pesos

- Duración del aprendizaje

- Observación durante las inyección de fallos de tipo SEU que la inversión de ciertos bits puede resultar en un mejor rendimiento de la red

La inversión de ese tipo de bits hace que se disminuya el tamaño del incremento y que se pueda acercar del mínimo vecino (local o global)

Por qué no explorar esta posibilidad? Universidad Complutense de Madrid, 17 Nov. 2015

- Inyecciones sucesivas de un SEU en todos los bits usados para memorizar los parámetros.

Conservación, para constituir el juego de pesos de la etapa siguiente, de la inversión del bit que resulta en el mayor aumento de prestaciones o, en caso de igualdad, del que conduce al menor error.
Condición de parada: cuando no hay ningún bit cuyo SEU aumenta las prestaciones de la red o disminuye el error

Etapa nº 0

Bit cuya inversión resulta en las mayores prestaciones Bit invertido conservado en una de las iteraciones precedentes

 Inyecciones sucesivas de un SEU er todos los bits usados para memorizar los parámetros.

Conservación, para constituir el juego de pesos de la etapa siguiente, de la inversión del bit que resulta en el mayor aumento de prestaciones o, en caso de igualdad, del que conduce al menor error.
Condición de parada: cuando no hay ningún bit cuyo SEU aumenta las prestaciones de la red o disminuye el error.

Etapa n° 0

Bit cuya inversión resulta en las mayores prestaciones Bit invertido conservado en una de las iteraciones precedentes

 Inyecciones sucesivas de un SEU er todos los bits usados para memorizar los parametros.

Conservación, para constituir el juego de pesos de la etapa siguiente, de la inversión del bit que resulta en el mayor aumento de prestaciones o, en caso de igualdad, del que conduce al menor error - Condición de parada: cuando no hay ningún bit cuyo SEU aumenta las prestaciones de la red o disminuye el



Etapa n° 1

error.

Bit cuya inversión resulta en las mayores prestaciones Bit invertido conservado en una de las iteraciones precedentes

Inyecciones sucesivas de un SEU er todos los bits usados para memorizar los parametros.

Conservación, para constituir el juego de pesos de la etapa siguiente, de la inversión del bit que resulta en el mayor aumento de prestaciones o, en caso de igualdad, del que conduce al menor error - Condición de parada: cuando no hay ningún bit cuyo SEU aumenta las prestaciones de la red o disminuye el



Etapa n° 2



error.

Bit cuya inversión resulta en las mayores prestaciones Bit invertido conservado en una de las iteraciones precedentes

Inyecciones sucesivas de un SEU er todos los bits usados para memorizar los parametros.

Conservación, para constituir el juego de pesos de la etapa siguiente, de la inversión del bit que resulta en el mayor aumento de prestaciones o, en caso de igualdad, del que conduce al menor error - Condición de parada: cuando no hay ningún bit cuyo SEU aumenta las prestaciones de la red o disminuye el



Etapa n° 3



error.

Bit cuya inversión resulta en las mayores prestaciones Bit invertido conservado en una de las iteraciones precedentes

- Inyecciones sucesivas de un SEU er todos los bits usados para memorizar los parametros.

Conservación, para constituir el juego de pesos de la etapa siguiente, de la inversión del bit que resulta en el mayor aumento de prestaciones o, en caso de igualdad, del que conduce al menor error.
Condición de parada: cuando no has ningún bit cuyo SEU aumenta las prestaciones de la red o disminuye el error.



Etapa nº N



Bit cuya inversión resulta en las mayores prestaciones

Bit invertido conservado en una de las iteraciones precedentes







Universidad Complutense de Madrid, 17 Nov. 2015

2,4% 77,2% 79,6%.

- En general, prestaciones
- Aumento de las prestaciones en la textura que se reconocía peor: la zona residencial.
- El nuevo conjunto de pesos es, al menos, tan robusto frente a los SEUs, que el conjunto inicial.
- El método estudiado fue aplicado por el CEA (Centro de Estudios Atómicos) a otra red neuronal y resultó en un aumento general de las prestaciones en un 2%.
- Mejora del método con « vuelta atrás » para evitar los mínimos locales.

Plan

- Introducción
- Efectos de las radiaciones en circuitos y sistemas integrados
- Redes de neuronas artificiales
- Problema estudiado « clasificación de texturas »
- El proyecto MPTB
- El « fuzzy control »
- Hopfield Neural Networks en FPGAs (HNNs)
- Conclusiones y perspectivas

El proyecto MPTB Microelectronic and Photonic TestBed

Motivaciones

- Reducción importante de los presupuestos militares al final de la guerra fría
- Demanda creciente de potencia de cálculo
- => Estudiar el comportamiento de componentes y tecnologías innovantes en entornos radiativos severos:
- aislantes
- memorias DRAM y SRAM en versiones comerciales y radhard
- convertidores A/D
- procesador RISC 3000 radhard
- redes neuronales

MPTB fue incluido en un satélite científico lanzado por NRL (Naval Research Labs) en 1997

Arquitectura adoptada para el experimento neuronal

Transputer T225

« Core » del sistema.
• Coprocesador L-Neuro 1.0 : circuito neuronal dedicado
• Memoria MHS 32 KB : Système.
• Memoria HITACHI 32 KB :

• PROMs :

Codigo del sistema, pesos /entrada de la red ANN • WatchDog :

Supervisor del sistema (detector de SEFIs) Sistema anti-Latchup :

Supervisa el consumo para detectar SELs



Experimento TIMA/CNES/CEA



Tarjetas en la carga útil



PANELA

PANEL B

PANEL C

Lanzamiento del satélite



MPTB siguió una órbita inclinada :

Altura al Apogeo: Altitura al Perigeo : Período : 38 811 km 1 326 km 713 min



Resultados: La dosis



Resultados : Tarjeta A7



Resultados : Tarjeta A7



79 upsets sur les paramètres

90 % de robustez

Resultados : Tarjeta B7



147 upsets sur les paramètres

91 % de robustez

Conclusiones sobre la la robustez de las redes neuronales

• Experiencias neuronales en MPTB totalemente operacionales y en vuelo desde 1997

upsets observados en los diferentes componentes upsets en las redes neuronales implementadas

• Robustez de las ANN implementadas demostrada

 Adquisición del « know how » respecto al desarrollo de experimentos a bordo de satélites, en particular para el « control inteligente »

Conclusiones sobre la la robustez de las redes neuronales

Resultados observados durante 5 años de operación del satélite

La redundancia de las ANNs las hace robustas frente a efectos de los SEUs en los datos de entrada y en los parámetros del ANN

Los datos observados en vuelo probaron que la robustez es similar para las dos tarjetas incluidas en el experimento MPTB: - 88% en caso de SEUs en las entradas del ANN (porción de la imagen estudiada)

- 82% en caso de SEUs perturbando los parámetros de la ANN

Algunos SEUs resultan en mejoras de la tasa de reconocimiento: de 77.5% a 82.5% (resp. 80%) en la tarjeta A7 (resp. B7). Recordemos que la tasa de reconocimiento en ausencia de errores es 77.5 % Universidad Complutense de Madrid, 17 Nov. 2015

Las dos memorias SRAM en las tarjetas



RST : RAM static Test CRC: Cyclic Redundancy Checksum MMI : Memory Mapped Interface

Errores detectados en las SRAMs

Período : finales de Diciembre 1997 a Febrero 1999

HITACHI Mem	MHS Mem
#SEU 119	149
Error rate (seu/bit/day) 1.45 E-5	7.25 E-5
# Double Bit Upsets 5	2.
Stuck-at-bit yes	no

La SRAM MHS es al menos 5 veces más sensible a los SEUs que la SRAM HITACHI

La SRAM MHS es sensible a errores múltiples en celdas de memoria adyacentes, situación previamente observada en los tests en aceleradores de partículas y en el experimento EXEW a bordo de la estación espacial

Errores detectados en las SRAMs

Tasas de errores estimadas y observadas (upset/bit/día)

SRAM	Tasa de error estimada	Tasa de error observada
Hitachi (HM62256)	4.8e-6 to 8e-6	1.45e-5
MHS (HM65756)	min. 1.3e-5 to 2e-5 max. 8.7e-5 to 1.4e-4	7.25e-5

Las predicciones basadas en los modelos estándar subestiman la tasa de error para la HM62256.

La tasa de error puede ser subestimada o sobre-estimada para la HM65756, pero la dispersión de los resultados obtenidos en experimentos con iones pesados no permite concluir.

Stuck-at bits

Este tipo de fallos es identificado cuando el mismo error (dirección y bit) es observado durante varios ciclos de lectura consecutivos

El macanismo que conduce a este tipo de fenómeno no es bien conocido Hipótesis:

- carga depositada en el gate oxide, que genera un sub-threshold leakage en un transistor y resulta que la celda queda « pegada » al estado de escritura

- interacción de una partícula enérgetica (ión pesado)
- dosis acumulada

Ejemplo ilustrativo: para el componente HM628128, los experimentos realizados con Co60 mostraron que comenzaban a ocurrir errores de tipo stuck-at-bit cuando la dosis alcanzaba unos 12 krad(Si). La combinación *total dose/heavy ions* reduce el nivel de dosis necesario para que este tipo de eventos suceda

Conclusiones sobre los test de las SRAMs

- Los resultados transmitidos por telemetría por MPTB muestran diferentes tipos de errores detectados en las memorias SRAM testeadas:
 - Single y Multiple Upsets
 - Stuck-at Bits
- La comparación de los datos obtenidos en vuelo con las tasas de errores predichas pone en evidencia desviaciones: *under or over estimations*.

Sitio Internet


Robustez de la lógica difusa

- Teoría :

Concepto de « agregados » 뻐 sistemas difusos robustos

• Práctica

Variaciones altamente no lineales de los controladores robustos

Primer estudio de la robustez de una aplicación espacial de un controlador difuso

Problema tratado

imágenes

Control desde la Tierra de las ruedas de un rover marciano a partir de imágenes estereoscópicas enviadas por un « lander » Señales de comando

聖

comandos

blanco



Resultados de la inyección de errores

Metodología :

- Inversión de un bit del micro-code del procesador WARP,
- Cargar en las memorias internas el micro-code con el error
- Cálculo de las salidas para el conjunto de motivos de entrada escogido
- Comparación de las salidas a los valores de referencia esperados

Tres tipos de comportamientos del WARP

- Valores de salida correctos para el conjunto de motivos de entrada escogidos (error de tipo *A*),
- Al menos una respuesta errónea para los motivos de entrada testeados (error de *tipo B*),
- No se obtiene respuesta del WARP durante cierto intervalo (*error de* tipo *C*).

niversidad Complutense de Madrid

Conclusiones sobre la robustez del control difusa

Buenas prestaciones frente a los SEUs

• Un experimento (colaboración TIMA/CNES/NASA) incluyendo el WARP y el modelo tratado fue instalado en la carga útil del satélite STRV (Space Technology Research Vehicle):

• Lanzamiento exitoso a fines de 1999 pero un problema de comunicación con los experimentos a bordo no permitió la obtención de datos observados en vuelo

Plan

- Introducción
- Efectos de las radiaciones en circuitos y sistemas integrados
- Redes de neuronas artificiales
- Problema estudiado « clasificación de texturas »
- El proyecto MPTB
- El « fuzzy control »
- Hopfield Neural Networks en FPGAs (HNNs)
- Conclusiones y perspectivas

A fault tolerant implementation of an HNN



A fault tolerant implementation of an HNN (cnt' d) Implemented fault tolerant calculations for a 4 neighbour nodes



A fault tolerant implementation of an HNN (cnt'd)

Architecture of the standard Serial Node



A fault tolerant implementation of an HNN (cnt' d) The proposed Fault Tolerant Serial Node (FTSN)



Fault Tolerant Serial Node is responsible for the multiplication of the synaptic weights by the input patterns, and the addition of the result with a serial input coming from another *FTSN*.

Fault injection performed on HNNs at netlist level **c) Experimental results obtained with NETFI** Application: Recognition of A, C & H characters (32-bits patterns)



HW overhead

Туре	Fault Tolerant RNH	Standard RNH	
Maximum frequency	104.811 MHz	157.604 MHz	
# Slices	6210 (33%)	2803 (15%)	
# Slice flip flop	8325 (22%)	3449 (9%)	
# 4 input LUTs	10593 (28%)	4800 (13%)	

Golden Run using NETFI (NETlist Fault Injection)

- 17 cycles required for the RNH
- 21 cycles required for the RNH-FT

A fault tolerant implementation of an HNN (cnt'd)

The Master Node (MN)



- For the fault tolerant design, n * log2 (n) master nodes are required for a row of "n" nodes.

- In first step, the MN is used to add two *FTSN* (fault tolerant serial nodes) whereas in following steps it is used to add two MNs that are distant of 2*n.

A fault tolerant implementation of an HNN (cnt' d) Architecture of a 8-nodes row



A row of "n" cells requires (n2-n) MNs:

- FTSNs are always used in the first step
- MN are used in the following steps.

Fault injection performed on HNNs at netlist level a. NETFI: NETlist Fault Injection

- Automated method to emulate faults using SRAM-based FPGAs

- Based on the netlist manipulation: replacement of Xilinx built-in devices by others having same functionalities and allowing fault injection
- Allows injecting various types of faults: SEU, SET and Stuck_at
- No restriction of the size of the FPGA in which the circuit will be emulated
- Can be applied to any digital circuit



Requirments:

- HDL code
- A FPGA platform
- APC
- "Synplify Pro" synthesis tool
- A place &root Xilinx tool
- Impact Xilinx tool for programming the FPGA
- A Linux-Like tool such as Cygwin
- MODNET (MODify NETlist)

The NETFI approach

For SEU injection:

- Flip-Flops (with and without Enable)
- Bloc Rams (BRAMS)
- Shift Registers SRL_16, RAMX1D..

For SET and Stuck_at fault injection:

- LUT (Look-Up Table)
- Porte Logiques (AND2, OR2, XOR2, MUXF5..)













ASTERICS: A generic and flexible test platform

- Design based on two FPGAs Xilinx Virtex 4
- Allows performing radiation ground testing & fault injection campaigns
- Test campaigns can be controlled by an Ethernet Gigabit
- Test platform monitored by the PowerPC embedded in the Control FPGA
- 180 I/O to interface a device under test (DUT)
- The DUT has access to 32Mb of SRAM and 512Mb of DDR-SDRAM







Various fault injection runs were performed:

-Only one fault injected per run

- Results were classified as:

- Silent: the fault injected has no effect.
- Error: the output of the HNN is not the expected one.
- Timeout: the HNN does not conclude.
- Converges: the HNN provides correct results after the expected running limit.

- The RNH netlist was modified to allow fault injection

- Different types of transient and permanent faults were injected

HNN original	Fault type	# Inj. Faults	Results Errors	Timeouts	Converges
	SEU	81831	79(0.096%)	1658(2.026%)	10865(13.277%)
	SET	125676	110(0.087%)	2623(2.087%)	15195(12.091%)
	Stuck at o	190509	9566(5.021%)	8417(4.418%)	19393(10.179%)
	Stuck at 1	98316	5444(5.537%)	3076(3.128%)	7659(7.790%)
HNN-FT	Fault type	# Inj. Faults	Results Errors	Timeouts	Converges
	SEU	137801	20(0.014%)	6(0.004%)	4789(3.475%)
	SET	149198	44(0.029%)	4(0.002%)	4193(2.810%)
	Stuck at 0	140516	3897(2.773%)	557(0.396%)	1667(1.186%)
	Stuck at 1	151803	4318(2.844%)	337(0.222%)	2486(1.637%)

Output Error

Timeout Errors



RNH output errors

Timeout Errors

- Experimental results show a significant improvement of fault tolerance
- A deep analysis of faulty results show that they are issued from faults in the RNH state machine.

Fault injection performed on the FPGA configuration memory

- Target HNN: Recognition of A, C & H characters (32-bits patterns)
- Used tool: NESSY (Non Intrusive Error injection System)
 Developed at Universidad Complutense de Madrid
- Testbench: recognition of a perturbed version of A pattern
- SEUs injected in all configuration bits used to implement the HNNs

Goal: analysis of the impact of SEUs in the configuration SRAM

Fault injection performed on the FPGA configuration memory

- Results obtained for the standard and the fault-tolerant HNN versions

- -Exhaustive fault injection performed in used SRAM configuration bits
- Each SEU injection takes 98,13 ns

- Massive fault injection required t 13 hours and 49 minutes hours for the HNN and 55 hours 17 minutes for the FT-HNN

Circuit	# Inj. SEU	Results Errors	Timeouts	Converges
HNN	3,573,760	25,172 (0.704%)	97,120 (2.718%)	874 (0.024%)
FT-HNN	12,495,040	55,224 (0.386%)	3,809 (0,027%)	7,540 (0.053%)

Robustness increases of a factor higher than 2

- -Timeouts decrease of a factor of 100
- Convergences require an average of 44 extra cycles

Conclusions

A fault tolerant implementation of Hopfield Neural Networks was explored

- Two versions of a RNH benchmark were implemented in an FPGA
- Fault injection campaigns were performed both at netlist level and at FPGA SRAM-configuration level
- Experimental results put in evidence a significant improvment of fault tolerance for the HNN hardened version

Perspectives

Apply the studied approach to a large RNH: a realistic application

Implement the fault tolerant version at hardware level

Validate its robustness by performing accelerated radiation tests

This FT RNH will candidate to be in the payload of a scientific nano-satellite. Collaboration with Brazil: Santa Maria University and SMDH (Santa Maria Design House)

Scientific production

This work is the result of a cooperation between Univ. of Balamand, Univ. Libanese, TIMA Lab and Univ. Complutense of Madrid

It lead to apublication accepted as a letter in Neurocomputig journal:

J. A. Clemente, W. Mansour, R. Ayoubi, F. Serrano, H. Mecha, H. Ziade, W. El Falou and R. Velazco, "Hardware Implementation of a fault-tolerant Hopfield Neural Network on FPGAs", in Neurocomputing, Vol. 171, pp. 1606-1609, January 2016.

Thanks for your attention and questions

