

MOTOR DE INFERENȚĂ GENERIC CU ARHITECTURĂ RECONFIGURABILĂ PENTRU SISTEME AUTONOME DE LUARE A DECIZIILOR

Viorel CĂRBUNE, Constantin ERHAN

Universitatea Tehnică a Moldovei

Abstract: În articol este propus spre realizare modelul unui motor de inferență cu arhitectură reconfigurabilă. Pentru aceasta s-au analizat o mulțime de soluții clasice existente și s-au propus soluții noi pentru soluționarea unei clase de probleme specifice. Pe lângă aceasta au fost propuse soluții ingineresti pentru rezolvarea problemelor de îmbunătățire și adaptare a soluțiilor clasice la specificul clasei de probleme.

Ca rezultate se pot menționa : modelul unui sistem fuzzy reconfigurabil care include arhitecturi hardware (fuzzyficator și defuzzyficator cu funcții de apartenență definite tabelar, motor de inferență reconfigurabil în baza celulelor logice fuzzy) pentru sisteme fuzzy reconfigurabile și diagramele de simulare care reflectă modul de funcționare a arhitecturilor utilizate cât și a modelului propus.

Cuvinte cheie: motor de inferență generic, arhitecturi reconfigurabile, sistem autonom, algoritm de luare a deciziilor.

Introducere:

Conceptul de motor de inferență generic se cere definit în cazul soluționării unor probleme specifice de luare automată a deciziilor, algoritmi cărora se pot modifica dinamic în timp. De obicei aceste sisteme se caracterizează prin capacitatea de autoorganizare a procesului decizional ceea ce face anevoioasă proiectarea unui asemenea sistem, cât și implementarea unui asemenea algoritm de instruire [1]. Arhitecturile propuse pentru soluționarea acestei clase de probleme pot fi utilizate atât pentru imlementarea algoritmilor generalizați de luare a deciziilor cât și a acelor dinamici dând posibilitate sistemului de a-și reconfigura algoritmul de luare a deciziilor în dependență de careva factori externi.

O altă trăsătură a metodologiei de dezvoltare a structurilor propuse este simplificarea imlementării algoritmilor de luare a deciziilor în baza structurilor hardware la etapa de proiectare. Avantajele acestei caracteristici se evidențiază la etapele de testare a sistemelor făcând posibilă racordarea acestora la noile cerințe de proiectare. Eficiența utilizării arhitecturilor elaborate este reflectată la etapa de proiectare a sistemelor de luare a deciziilor cu motor de inferență dinamic sau reconfigurabil.

Motor de inferență generic

Logica vagă a suferit o răspîndire largă în domeniile în care metodele clasice de rezolvare a problemelor nu aduc rezultate scontate. Aceasta este utilizată la descrierea proceselor care manifestă o dependență neliniară sau chiar probabilistică dintre variabilele care descriu procesul în cauză. Logica fuzzy se mai folosește și la optimizarea algoritmilor obținuți prin metode clasice.

Una din metodele de implementare a logicii fuzzy în hardware ar fi utilizarea funcțiilor de apartenență definite tabelar la proiectarea fuzzyficatorului [2]. Această metodă presupune utilizarea circuitelor de memorie, care dau posibilitatea de a defini funcții de apartenență neliniare, diferite de cele clasice (triunghiulare, trapezoidale...) fapt ce deseori în practică este mai convenabil. Funcțiile de apartenență determinate automat prin metode numerice, adesea nu pot fi definite liniar fără a se introduce careva rată de eroare adăugătoare la etapa de interpolare.

O altă metodă utilizată frecvent la generalizarea sistemelor fuzzy este utilizarea regulilor de inferență configurabile care oferă un avantaj față de cele predefinite deoarece pot fi modificate în timp real. Utilizarea structurilor configurabile față de procesoarele Fuzzy specializate, fie ele chiar și de performanță, exclude etapa de proiectare a circuitelor integrate de fiecare data cînd se dorește modificarea structurii interne a sistemului, exclude legatura între numărul de intrări și proiectarea unui nou nucleu de processor. Această problemă poate fi ușor soluționată prin utilizarea circuitelor FPGA care au un număr mare de porturi de intrare/ieșire fiind necesară doar reassignarea pinilor [4]. Chiar și proiectarea unui nou nucleu fuzzy se simplifică semnificativ prin utilizarea bibliotecilor de elemente logice fuzzy.

Conceptul de motor de inferență reconfigurabil poate servi pentru dezvoltarea automată a unui algoritm de luare a deciziilor implementat inițial cu condiția insuficienței datelor inițiale [4]. Prin urmare, este de așteptat ca utilizarea motoarelor de inferență reconfigurabile să dea posibilitatea de implementare a diferitor algoritmi de învățare a sistemelor automate de luare a deciziilor și astfel să se apropie de procesul uman de luare a deciziilor și de inteligența acestuia.

Pentru implementarea regulilor de inferență în hardware s-a proiectat un set de elemente logice fuzzy după analogia celor din logica clasică [3]. Astfel s-a implementat funcția operatorului logic Fuzzy- AND realizat pe bază de comparator/multiplexor care de facto are două intrări și o ieșire pe n biți și selectează valoarea minimă dintre cele două intrări.

Funcția operatorului logic Fuzzy-OR s-a realizat pe bază de comparator/multiplexor care are două intrări și o ieșire pe n biți și selectează valoarea maximă dintre cele două intrări [3].

În dependență de complexitatea regulilor de inferență trebuie de ținut cont de întârzierile care pot apărea în aceste blocuri și de frecvența de lucru a sistemului care se alcătuiește din întârzierea maximă. În acest caz se exclude apariția coleziunilor la ieșirile motorului de inferență.

După analogia cu PLA (Matrici Logice Programabile) s-a realizat structura motorului de inferență reconfigurabil sau generic, care prin analogie realizează toate combinațiile conjunctive posibile între calificativele variabilelor de intrare. La urmatorul pas acestea sunt aplicate la blocul de multiplexoare pentru a crea totalitatea disjuncțiilor definite de tabelul de adevăr. Pentru a putea varia numărul de conjuncții din forma disjunctivă a regulii de inferență s-a introdus semnalul MIN care poate anula întreaga construcție conjunctivă. Pentru realizarea funcțiilor constante, de asemenea pot fi utilizate semnalele MIN la toți termenii conjunctivi sau aplicarea semnalului MAX la primul termen conjunctiv, ceea ce va duce la ignorarea valorilor celorlalți termeni.

Utilizarea acestor tipuri de elemente face ca orice regulă de inferență să fie descrisă sub forma :

$$C_{Cal_i} = \sum_{k=1}^n Op_k;$$

Unde C_{Cal_i} este variabila C determinată de calificativul i, iar

$$Op_1 = \begin{cases} MIN, & PR[1..0] = '00' \\ MAX, & PR[1..0] = '01' \\ A_{Cal1} \& B_{Cal1}, & PR[1..0] = '10' \end{cases};$$

$$Op_1 = Var_1 Cal_1 \& Var_2 Cal_1 \& \dots \& Var_{N-1} Cal_1 \& Var_N Cal_1;$$

$$Op_2 = Var_1 Cal_1 \& Var_2 Cal_1 \& \dots \& Var_{N-1} Cal_1 \& Var_N Cal_2;$$

□

$$Op_M = Var_1 Cal_1 \& Var_2 Cal_1 \& \dots \& Var_{N-1} Cal_1 \& Var_N Cal_M;$$

□

$$Op_{M*N-1} = Var_1 Cal_M \& Var_2 Cal_M \& \dots \& Var_{N-1} Cal_M \& Var_N Cal_{M-1};$$

$$Op_{M*N} = Var_1 Cal_M \& Var_2 Cal_M \& \dots \& Var_{N-1} Cal_M \& Var_N Cal_M;$$

Pentru cazul cu 2 variabile de intrare și o variabilă de ieșire determinate de două calificative fiecare, regulile de inferență capătă forma:

$$C_{Cal_i} = Op_1 \vee Op_2 \vee Op_3 \vee Op_4$$

unde :

$$Op_1 = \begin{cases} MIN, & PR[1..0] = '00' \\ MAX, & PR[1..0] = '01' \\ A_{Cal1} \& B_{Cal1}, & PR[1..0] = '10' \end{cases};$$

$$Op_2 = \begin{cases} MIN, & PR[2] = '0' \\ A_{Cal1} \& B_{Cal2}, & PR[2] = '1' \end{cases};$$

$$Op_3 = \begin{cases} MIN, & PR[3] = '0' \\ A_{Cat2} \& B_{Cat1}, & PR[3] = '1' \end{cases};$$

$$Op_4 = \begin{cases} MIN, & PR[4] = '0' \\ A_{Cat1} \& B_{Cat2}, & PR[4] = '1' \end{cases};$$

Structura și funcționarea modelului prezentat:

Pentru a se sublinia utilitatea modelului propus, s-a recurs la prezentarea sistemului fuzzy integral ceea ce oferă o viziune mai completă asupra eficienței utilizării acestui model cât și asupra avantajului utilizării bibliotecii de elemente logice fuzzy la etapa de proiectare.

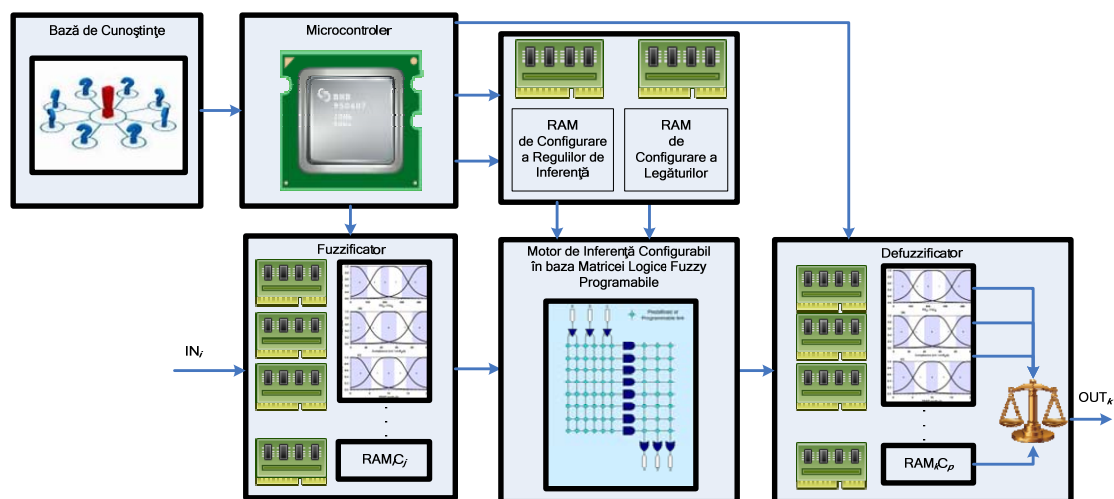


Fig. 1 Schema de structură a sistemului de luare automată a deciziilor cu motor de inferență generic

Ca elemente componente se pot defini:

Bază de Cunoștințe – Reprezintă o structură tabelară, relațională în care se păstrează funcțiile de apartenență, programele de configurare a regulilor de inferență și funcțiile de defuzificare. La cererea microcontrolerului datele, conform algoritmului de selectare, se încarcă în blocurile respective de RAM pentru a configura sistemul fuzzy.

Microcontroler – Reprezintă un microprocesor clasic destinația căruia este de a selecta și extrage din baza de cunoștințe funcțiile de apartenență pentru variabilele fuzzy, regulile de inferență pentru motorul de inferență și metoda de defuzificare pentru defuzificator. Acesta funcționează în conformitate cu algoritmul stocat în memoria de program.

Fuzzificator – Reprezintă blocuri de RAM în care tabelar se definesc funcțiile de apartenență pentru fiecare calificativ al variabilelor fuzzy.

Motor de Inferență Configurabil – Motor de inferență configurabil realizat după analogia cu PLA, elementele structurale reprezentând matrici de elemente logice fuzzy AND și OR (Programable Fuzzy Logic Array).

RAM de Configurare a Regulilor de Inferență – bloc de memorie operativă destinat păstrării datelor de configurare pentru regulile de inferență. În dependență de această configurare, motorul de inferență poate implementa una din mulțimea de funcții definită de numărul de variabile de intrare.

RAM de Configurare a Legăturilor – bloc de memorie operativă destinat păstrării datelor de configurare pentru legăturile dintre elementele logice fuzzy. În dependență de această configurare, motorul de inferență poate stabili careva legături între elementele structurii sale interne.

Fuzzy AND Array – matrice de elemente logice fuzzy AND care realizează funcția MIN;

Fuzzy OR Array – matrice de elemente logice fuzzy OR care realizează funcția MAX;

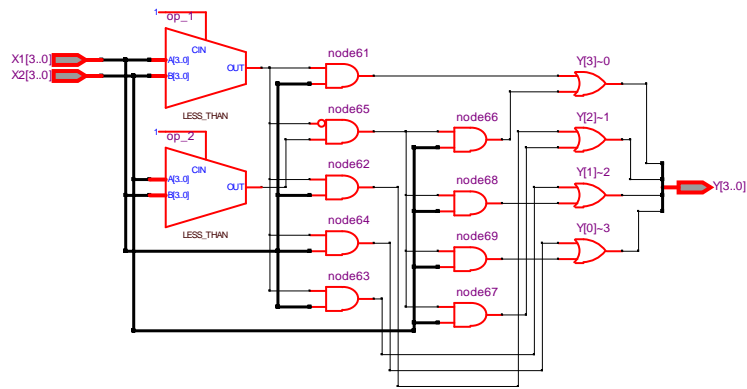


Fig. 2 Schema logică pentru elementul Fuzzy AND

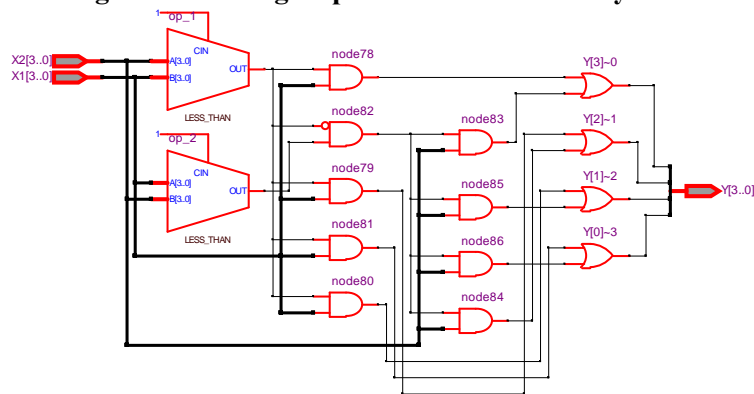


Fig. 3 Schema logică pentru elementul Fuzzy OR

Defuzzificator – Reprezintă blocuri de RAM în care tabelar se definesc funcțiile de defuzzificare pentru fiecare variabilă fuzzy.

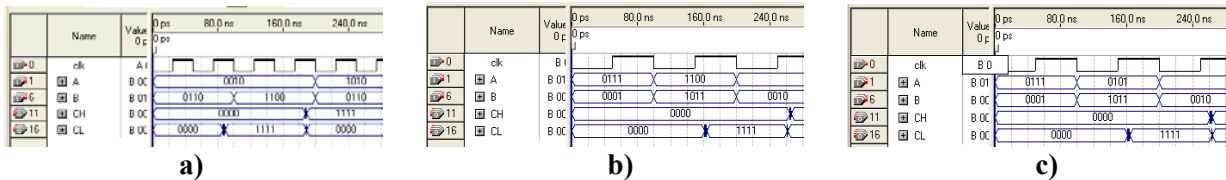


Fig. 4 Diagramele de timp pentru Funcțiile a) XOR, b) AND, c) OR

Concluzii: Structura sistemului integru devine inevitabil mai complexă deoarece înglobează o funcționalitate extinsă, însă procesul de proiectare se simplifică substanțial. Utilizarea practică a acestor motoare de inferență poate fi apreciată în procesul de testare a sistemelor automate de luare a deciziilor în cazul în care regulile de inferență necesită modificări pe parcursul etapei de proiectare. În acest caz algoritmul decizional poate fi ușor modificat fără a se reproiecta și fără a se modifica structura internă a sistemului, făcându-l astfel destul de flexibil și universal în limitele aceleiași clase de probleme.

Bibliografie:

1. Marek Karkula, Lech Bukowski. „Integration of Discrete Event Simulation, Decision Tables and Fuzzy System in Modelling of Logistics Processes”. In: Adam Grzech, Pawel Swiatek, Jaroslaw Drapala (Eds). *Advances in Systems Science*, Warsaw, 2010: 113–121.
2. Marco Rosso, Luigi Caponetto. „Hardware Implementation of Intelligent Systems”. In: H. Teodorescu, L. Jain, A. Kandel (Eds). *Hardware Implementation of Intelligent Systems*. Heidelberg, NY: Physica-Verlag 2001: 91–116.
3. Gian Carlo Cardarilli, Roberto Lojacono, Marco Re. „High Performance Fuzzy Processors”. In: H. Teodorescu, L. Jain, A. Kandel (Eds). *Hardware Implementation of Intelligent Systems*. Heidelberg, NY: Physica-Verlag 2001: 121–145.
4. Davide Falchieri, Alessandro Gabrielli, Enzo Gandolfi. „A Digital Fuzzy Processor for Fuzzy-Rule-Based Systems”. In: H. Teodorescu, L. Jain, A. Kandel (Eds). *Hardware Implementation of Intelligent Systems*. Heidelberg, NY: Physica-Verlag 2001: 147–163.