

Universitatea „Petru Maior” din Târgu Mureș
Facultatea de Inginerie Electrică și Calculatoare
Master MSE I – Managementul Sistemelor Energetice

Sisteme SCADA
Schema funcțională a unei centrale nucleare
- Examen –

Masterand:
Muntean Raul

Cuprins

1	Introducere:	3
2	Interfața HMI:	4
3	Structurarea codului și rularea programului:	5
4	Anexe	8
4.1	funcția background și aar	8
4.2	Funcția sursa1	9
4.3	Funcția sursa2	9
4.4	Funcția sursa3	9
4.5	Funcția intarziere	10
4.6	Funcția pompapa	10
4.7	Funcția pompacondensare.....	11
4.8	Funcția generatorelectri	11
4.9	Funcția controlrods	11
4.10	Funcția reactienucleara	11
4.11	Funcția failsafe.....	12

1 Introducere:

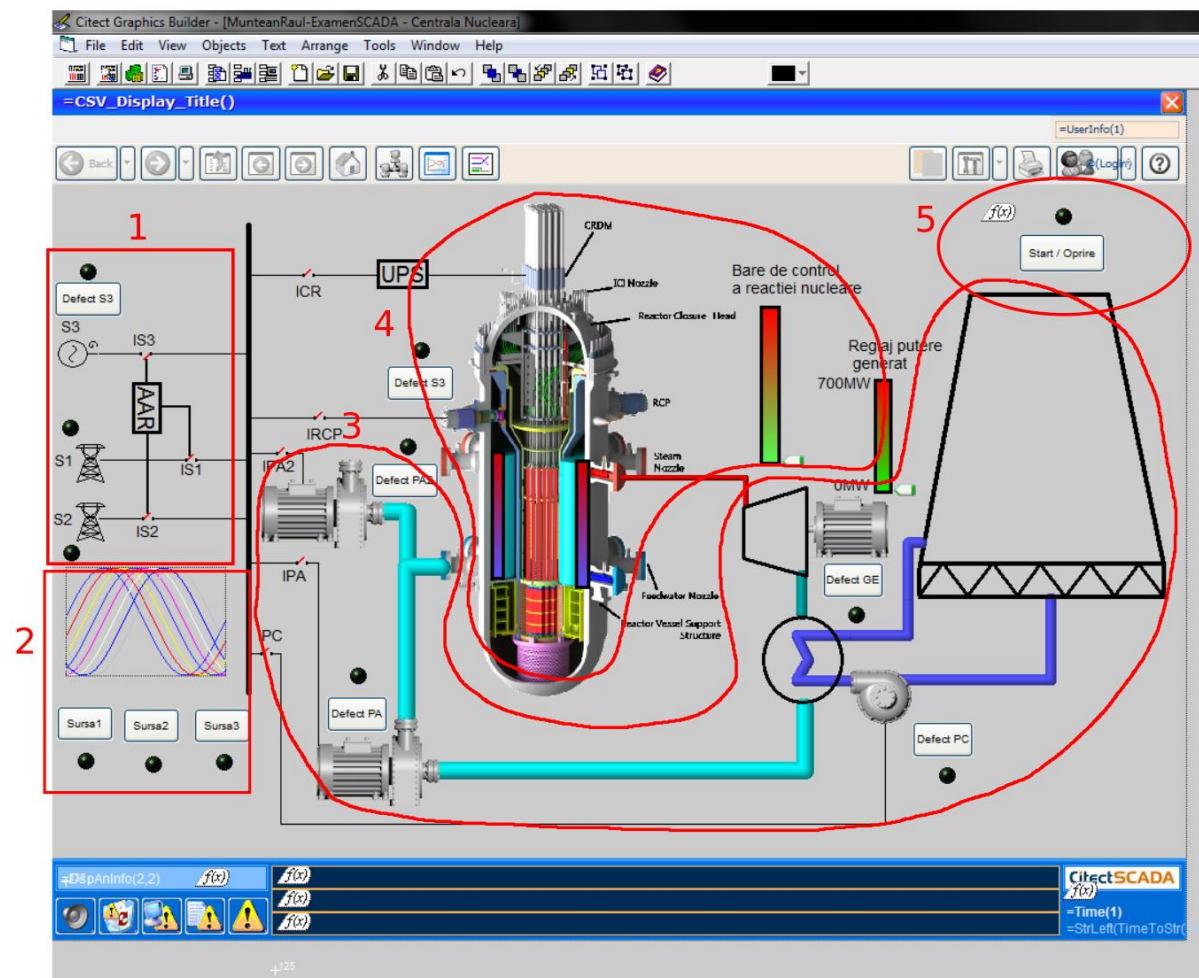
În această lucrare s-a urmărit realizarea unei interfețe HMI (human machine interface) pentru monitorizarea și controlul unei scheme funcționale a unei centrale nucleare.

La bază o centrală nucleară folosește căldura emisă în urma reacției nucleare pentru a produce abur. Curentul electric se obține prin antrenare axului unui generator electric de către o turbină acționată de aburul produs. Urmând această idee programul de față are la bază monitorizarea temperaturii și a oricărui defect ce poate duce la creșterea acestuia peste limita admisă.

O centrală nucleară din punctul de vedere al alimentării cu energie electrică intră în categoria de consumator zero (cu potențial de pierderi de vieți omenești), de aceea schema funcțională realizată are trei surse de alimentare dintre care: două sunt reprezentate de rețele electrice separate și o a treia reprezentată de un generator electric. În schemă mai apare un UPS care are rolul alimentării mecanismului de conducere a barelor de control și pentru această lucrare l-am considerat ca fiind funcțional în permanență. Comutarea între surse se face prin intermediul unui AAR ce permite schimbarea între sursele de alimentare sigură prin implementare unei întârzieri, iar întrerupătoarele reprezentate în schemă sunt contactoare sau întrerupătoare capabile de o comutare în sarcină.

Circuitul apă – abur – turbină – condensator este identic cu cel al centralelor electrice pe combustibili fosili și este alcătuit din două pompe de circulație, una fiind de rezervă, și o pompă pentru circulația apei prin condensator și turnul de răcire.

2 Interfața HMI:

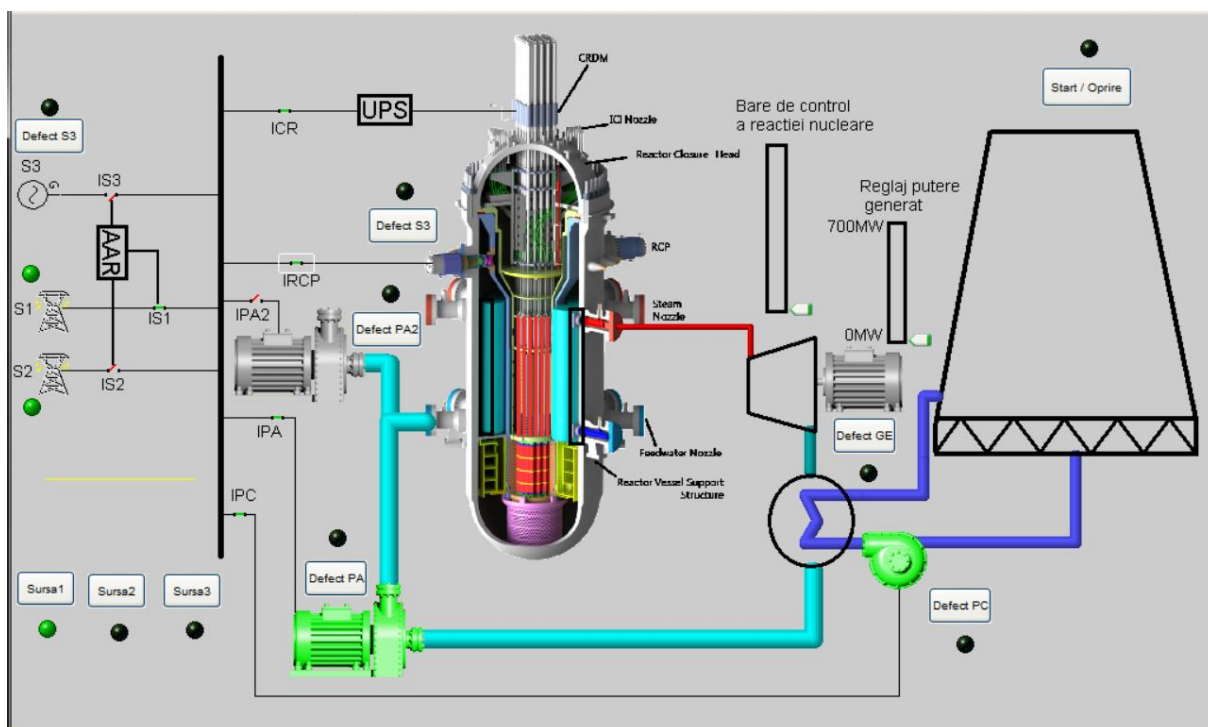


- 1. Segmentul de alimentare din cele trei surse s1, s2 și s3 prin intermediul unui AAR care controlează cele trei întrerupătoare IS1, IS2 și IS3. Pornirea și oprirea acestora se poate face apăsând iconița fiecăruia.
- 2 Segmentul de afișare a curbelor de tensiune a surselor, alegerea afișării unei surse se face prin intermediul butoanelor Sursa1, Sursa 2 și Sursa 3.
- 3 Segmentul de circuit apă – abur – turbină – condensator ce conține cele două pompe de circulație a apei și pompa condensatorului.
- 4 Segmentul reactorului nuclear unde se află indicatoarele de temperatură a reactorului nuclear, indicatorul de nivel a barelor de control și sliderul de control a puterii care se dorește a fii generată.
- 5 Reprezintă butonul de start și oprire a programului prin intermediul căruia se aduce programul înapoi la starea inițială.
- Butoanele intitulate defect au rolul de a simula defecte ale echipamentelor.

3 Structurarea codului și rularea programului:

Pentru a obține o funcționare neîntreuptă a programului s-a folosit o funcție de fundal intitulată **background()** care va fi chemată la fiecare 0.25s. În această funcție în starea zero a etichetei **kill** programul setează parametrii la starea inițială altfel incrementează o etichetă contor **i** (poate fi considerată un ceas al programului) și cheamă funcțiile: **aar**, **sursa1**, **sursa2**, **sursa3**, **pompaapa**, **pompacondensare**, **controlrods**, **reactienucleara**, **generatorelectric**, **failsafe**.

Stare inițială se poate vedea în figura următoare:



Funcțiile **sursa1** și **sursa2** au rolul de a verifica starea sursei dacă este pornită sau nu și în funcție de aceasta apare sau nu tensiune pe linie și la dispariția tensiunii automat se deconectează întrerupătorul acelei surse.

Funcția **sursa3** verifică starea de defect al echipamentului, starea celorlalte surse și a steagului propriu **steags3**. Dacă există tensiune pe una dintre celelalte surse aceasta nu va porni și nici nu va acționa întrerupătorul **is3**. Nevoia de a-și verifica propriul steag este dat de faptul că un generator nu pornește imediat ci are un timp de pornire.

Funcția **aar** conține deciziile de alegere a cărei surse de alimentare va fi conectată. În cazul nevoii de alimentare de la sursa 3 această funcție cheamă funcția **întarziere** cu argumentele **tip** și **val** după care se verifică eticheta de finalizare a întârzierii **steagDelay[tip]** dacă aceasta este adevărată eticheta sursei 3 va lua valoarea adevărat.

Funcția **întarziere** folosește argumentul **tip** pentru a crea o categorii de întârzieri diferite una de alta iar argumentul **val** este folosit pentru a specifica timpul de întârziere. În cazul sursei trei a luat valoare de 30 de incrementări ale conotrului **i**. De-a lungul programului am ajuns să folosesc 15 întârzieri diferite cu timpi diferiți având ca rezultat un program care nu se întrerupe pe sine în bucle for.

Funcția **pompaapa** are un rol asemănător cu cel al funcției **aar** și anume alege în funcție de defecte care pompă este alimentă cu energie electrică.

Funcția **pompacondensare** are simplul rol de a verifica starea de defect a acesteia.

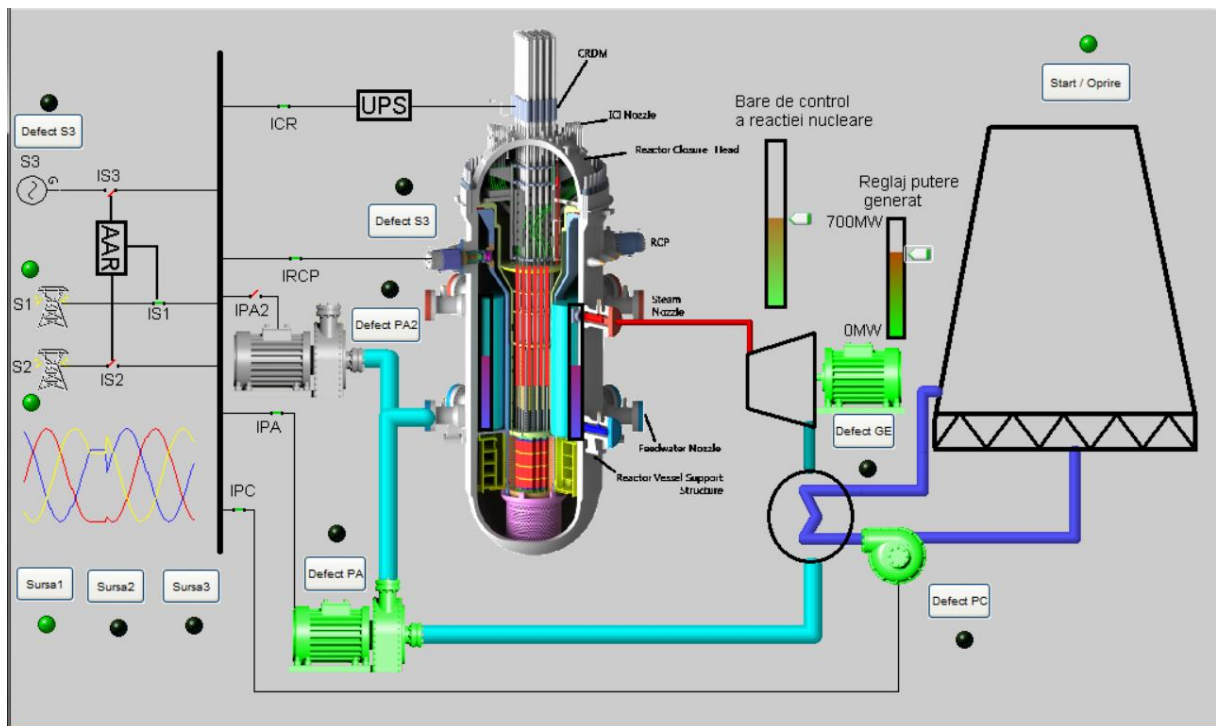
Funcția **generatorelectric** verifică nivelul dorit de putere ales prin intermediul slider-ului și pornește generatorul cu o întârziere de 10 incrementări **i**.

Funcția **controlrods** este analog cu funcția de mai sus doar că după întârziere are loc o incrementare sau decrementare a nivelului barelor de control.

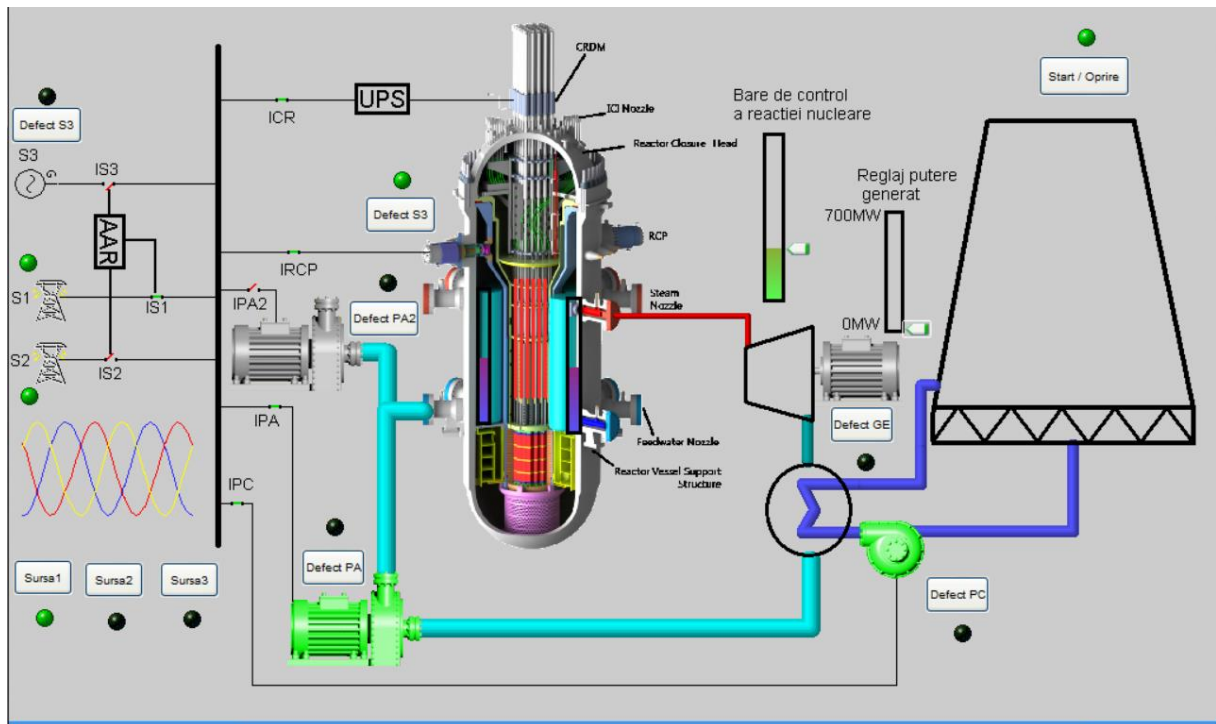
Funcția **reactienucleara** simulează creșterea sau scăderea temperaturii mai rapide sau mai lente în funcție de nivelul barelor de control. În realitate nivelul barelor de control opresc sau măresc frecvența reacțiilor nucleare ducând la creșterea sau scăderea temperaturii. Metoda de realizare este prin folosirea funcției **întarziere** și a incrementării etichetei **timp**.

Ultima funcție fiind funcția **failsafe** folosește instrucțiunile decizionale pentru a da valoarea zero a etichetei **putere** care în schimb oprește centrala nucleară. Aici este implementată și protecția la supraîncălzire prin faptul că dacă eticheta **temp** are o valoare de peste 90 timp de 50 de incrementări **i** atunci valoarea etichetei **temp** devine 560 adică echivalentul la 80% din puterea maximă generată.

În figura de mai jos se poate observa funcționarea normală a centralei nucleare. În cazul acesta barele de control nu au fost scoase îndeajuns pentru a atinge puterea cerută, iar temperatura încă va crește.



În figura următoare se poate observa că la apariția unui defect major cum ar fi oprirea pompelor de racire a reactorului programul va seta automate eticheta **putere** pe valoarea zero urmând ca barele de control și temperatura sa scadă în nivel.



4 Anexe

4.1 funcția background și aar

```
GLOBAL INT delay[20];
GLOBAL INT steagDelay[20];
GLOBAL INT j;

FUNCTION background()
    IF kill=0 THEN
        i=0;
        steags1=1;
        steags2=1;
        steags3=0;
        grafics1=1;
        grafics2=0;
        grafics3=0;
        IS1=1;
        IS2=0;
        IS3=0;
        icr=1;
        ircp=1;
        ipa=1;
        ipa2=0;
        ipc=1;
        PA=1;
        PA2=0;
        PC=1;
        RCP=0;
        putere=0;
        temp=0;
        crdm=0;
        WHILE j<21 DO
            j=j+1;
            delay[j]=0;
            afisaredelay[j]=0;
        END
    ELSE
        j=0;
        i=i+1;
        aar();
        sursa1();
        sursa2();
        sursa3();
        pompaapa();
        pompacondensare();
        controlrods();
        reactienucleara();
        generatorelectric();
        failsafe();
    END
END

FUNCTION aar();

    IF (steags1 = 1 OR steags2 = 1) AND (steags1 = 1 OR steags3 = 1)
    THEN
        IF steagDelay[1] = 1 THEN
            is1 = 1;
        END
    ELSE
        IS1=0;
    END
END
```



```

IF steags1 = 0 OR steags3 = 1 THEN
    IF steags2 = 1 THEN
        IF steagDelay[2] = 1 THEN
            is2 = 1;
        END
    END
ELSE
    IS2 = 0;
END

IF steags1 = 0 AND steags2 = 0 THEN
    INTarziere(3,30);
    IF steags3 = 1 THEN
        is3 = 1;
    END
END
END

```

4.2 Funcția sursa1

```

FUNCTION sursa1()
    IF steags1 = 1 THEN
        s1[0]=400*Sin(2*Pi*50*i/5000);
        s1[1]=400*Sin(2*Pi*50*i/5000 - 2*Pi/3);
        s1[2]=400*Sin(2*Pi*50*i/5000 + 2*Pi/3);
        delay[2]=0;
        INTarziere(1,10);
    ELSE
        s1[0]=0;
        s1[1]=0;
        s1[2]=0;
        IS1=0;
        delay[1]=0;
        steagDelay[1]=0;
    END
END

```

4.3 Funcția sursa2

```

FUNCTION sursa2()
    IF steags2 = 1 THEN
        s2[0]=400*Sin(2*Pi*50*i/5000);
        s2[1]=400*Sin(2*Pi*50*i/5000 - 2*Pi/3);
        s2[2]=400*Sin(2*Pi*50*i/5000 + 2*Pi/3);
        INTarziere(2,10);
    ELSE
        s2[0]=0;
        s2[1]=0;
        s2[2]=0;
        delay[2]=0;
        IS2=0;
        steagDelay[2]=0;
    END
END

```

4.4 Funcția sursa3

```

FUNCTION sursa3()
//sursa de rezerva
    IF defects3=0 THEN
        IF steags1 = 1 OR steags2 = 1 THEN
            steags3=0;
            delay[3]=0;
            IS3=0;
        ELSE
            IF steagDelay[3] = 1 THEN

```

```

        steags3 = 1;
    END
END
IF steags3 = 1 THEN
    s3[0]=400*Sin(2*Pi*50*i/5000);
    s3[1]=400*Sin(2*Pi*50*i/5000 - 2*Pi/3);
    s3[2]=400*Sin(2*Pi*50*i/5000 + 2*Pi/3);
ELSE
    s3[0]=0;
    s3[1]=0;
    s3[2]=0;
    IS3=0;
END
ELSE
    s3[0]=0;
    s3[1]=0;
    s3[2]=0;
    IS3=0;
    steags3=0;
END
END

```

4.5 Funcția intarziere

```

FUNCTION INTarziere(INT tip,INT val)
delay[tip] = delay[tip] + 1 ;
afisaredelay[tip] = delay[tip];
    IF delay[tip] MOD val = 0 THEN
        steagDelay[tip] =1;
    ELSE
        steagDelay[tip]=0;
    END
END

```

4.6 Funcția pompapa

```

FUNCTION pompaapa ()
    IF (PA2=0 AND defectPA=0) AND (steags1 = 1 OR steags2 = 1 OR steags3 =
1) THEN
        INTarziere(4,10);
        IF steagDelay[4]=1 THEN
            ipa=1;
            PA=1;
        END
    ELSE
        PA=0;
        ipa=0;
    END

    IF (PA=0 AND defectPA2=0) AND (steags1 = 1 OR steags2 = 1 OR steags3 =
1) THEN
        INTarziere(4,10);
        IF steagDelay[4]=1 THEN
            ipa2=1;
            PA2=1;
        END
    ELSE
        PA2=0;
        ipa2=0;
    END

    IF defectPA=0 AND defectPA2=0 THEN
        PA2=0;
        ipa2=0;
    END

```

```

    INTarziere(4,10);
    IF steagDelay[4]=1 THEN
        ipa=1;
        PA1=1;
    END
END
END

```

4.7 Funcția pompacondensare

```

FUNCTION pompacondensare()
    IF defectPC=0 THEN
        PC=1;
        IF (steags1 = 1 OR steags2 = 1 OR steags3 = 1) THEN
            INTarziere(5,10)
            IF steagDelay[5]=1 THEN
                ipc=1;
            END
        END
    ELSE
        ipc=0;
        pc=0;
    END
END
END

```

4.8 Funcția generatorelectri

```

FUNCTION generatorelectric()

    IF putere > 0 THEN
        INTarziere(12,10)
        IF steagdelay[12]=1 THEN
            GE=1;
            delay[12]=0;
        END
    ELSE
        GE=0;
    END
END
END

```

4.9 Funcția controlrods

```

FUNCTION controlrods()

    IF crdm < putere/7 THEN
        INTarziere(13,2);
        IF steagDelay[13]=1 THEN
            crdm=crdm+1;
        END
    ELSE
        INTarziere(13,3);
        IF steagDelay[13]=1 THEN
            crdm=crdm-1;
        END
    END
END
END

```

4.10 Funcția reacțienucleara

```

FUNCTION reacțienucleara()
    IF crdm >= 0 AND crdm <= 28 THEN
        IF temp <= crdm THEN
            INTarziere(6,5);
            IF steagDelay[6]=1 THEN
                temp = temp + 1;
            END
        ELSE

```

```

        INTarziere(7,2);
        IF steagDelay[7]=1 THEN
            temp = temp - 1;
        END
    END
END

IF crdm > 28 AND crdm <= 55 THEN
    IF temp <= crdm THEN
        INTarziere(8,3);
        IF steagDelay[8]=1 THEN
            temp= temp + 2;
        END
    ELSE
        INTarziere(9,5)
        IF steagDelay[9]=1 THEN
            temp = temp - 1;
        END
    END
END

IF crdm > 55 THEN
    IF temp <= crdm THEN
        INTarziere(10,2)
        IF steagDelay[10]=1 THEN
            temp = temp + 3;
        END
    ELSE
        INTarziere(11,5)
        IF steagDelay[11]=1 THEN
            temp = temp -1;
        END
    END
END
END
END

```

4.11 Funcția failsafe

```

FUNCTION failsafe()

    IF (defectPA=1 AND defectPA2=1) OR (steags1=0 AND steags2=0 AND
steags3=0) THEN
        INTarziere(14,40);
        IF steagDelay[14]=1 THEN
            putere=0;
        END
    END

    IF PC=0 THEN
        putere=0;
    END

    IF defectGE=1 THEN
        GE=0;
        putere=0;
    END

    IF temp > 90 THEN
        INTarziere(15,50)
        IF steagDelay[15]=1 THEN
            putere = 560;
        END
    END

    IF RCP=1 OR ircp = 0 THEN

```

```
putere = 0;  
END  
END
```