

Cerințe generale pentru efectuarea temei suplimentare

Reguli de formatare și tehnoredactare

Documentația aferentă temei suplimentare va fi structurată pe capitole și va include următoarele elemente obligatorii:

- a. *Pagina de titlu*
- b. *Cuprins* – lucrarea va avea un cuprins care să conțină cel puțin titlurile tuturor capitolelor însotite de numărul paginii la care începe fiecare capitol
- c. *Lista figurilor și lista tabelelor* – în cazul în care documentația conține figuri (imagini, grafice), aceste vor fi prezentate, imediat după cuprins, sub forma unor liste (separat pentru figuri și tabele), care conțin numele fiecărui element și numărul paginii la care se află acesta;
- d. *Introducere* – aceasta va conține obiectivele generale ale proiectului, metodologia folosită și va rezuma contribuția studentului și rezultatele obținute
- e. *Capitole* – lucrarea va conține capituloane numerotate crescător (introducerea nu se numerotează, primul capitol fiind Capitolul 1)
- f. *Concluziile proiectului* – în această parte a lucrării se regăsesc cele mai importante concluzii din lucrare. Concluziile proiectului nu se numerotează ca și capitol.

Reguli de redactare a lucrării

Pentru redactare se va folosi un program de tehnoredactare (exemplu: Microsoft Word, LaTeX, Open Office). Formatul lucrării va fi A4, numărul de pagini (exclusiv anexe) fiind între 10 și 30, cu următoarele elemente:

- a. Limba de redactare: Se va utiliza obligatoriu limba programului de studiu al absolventului; pentru limba română, se impune utilizarea diacriticelor specifice (ă, î, â, ş, ţ).
- c. Marginile paginii – se vor utiliza următoarele valori pentru marginile paginii:
 - stânga: 2 cm
 - dreapta: 2 cm
 - sus: 2 cm
 - jos: 2 cm
- e. Spațiere între rânduri - textul va respecta o spațiere între rânduri de 1 - maxim 1.2 rânduri.
- f. Alinierea textului în cadrul paragrafelor - textul din cadrul paragrafelor normale va fi aliniat între marginile din stânga și dreapta (justified). Excepție fac titlurile capitolelor, care pot fi alinate centrata, precum și etichetele tabelelor și figurilor;
- g. Font – fontul utilizat pentru redactare (atât în capituloane, cât și în anexe):
 - Pentru textul lucrării se va utiliza un font de tip proporțional (exemplu: Times New Roman, Computer Modern etc), cu dimensiunea corpului de 11..12 puncte. Este recomandată folosirea unui font cu serifuri cum sunt cele exemplificate.

- Pentru listarea codului sursă se va utiliza un font cu dimensiune fixă (exemplu: Courier New etc), cu dimensiunea corpului de 8 .. 12 puncte. Este recomandabilă reducerea corpului în anexe. Listarea codului sursă se poate face pe 2 coloane.

h. Numerotarea paginilor - numerotarea paginilor se face consecutiv cu cifre arabe (1,2,...) începând cu pagina de titlu, până la ultima pagină a proiectului, inclusiv anexele, dar numărul paginii apare doar începând cu Introducerea. Numărul de pagină se inserează în subsolul paginii, centrat.

i. Antetul paginii – optional – poate apărea începând cu introducerea și poate conține elemente de identificare cum ar fi numele studentului, titlul capitolului etc.

j. Tabele – tabelele se numerotează cu 2 cifre, prima reprezentând numărul capitolului, iar cea de a doua reprezentând numărul tabelului din capitolul respectiv. Fiecare tabel are număr și titlu, care se menționează sub tabel, centrat.

k. Figuri - figurile (aici sunt incluse imagini, grafice, capturi de ecran) se numerotează cu 2 cifre, prima reprezentând numărul capitolului, iar cea de a doua fiind numărul figurii din capitolul respectiv; fiecare figură are număr și titlu, care se menționează sub figură, centrat.

Componența proiectului scris

Proiectul va include următoarele componente:

- Schema bloc a sistemului.
- Diagrama cazurilor de utilizare a sistemului.
- Descrierea funcționalității porturilor de intrare și ieșire.
- Proiectarea codului sursă corespunzător sistemului.
- Implementarea codului sursă.
- Diagramele structurale corespunzătoare codului sursă.
- Diagramele secvențiale corespunzătoare codului sursă.
- Testarea funcționalității sistemului.

DESCRIEREA TEMEI – Sistem de Sortare și Depozitare a Pachetelor pe o Bandă Transportoare

CERINȚĂ

Se va implementa un sistem automatizat de sortare și depozitare a pachetelor utilizând circuite logice combinaționale și secvențiale. Sistemul trebuie să fie capabil să recunoască și să redirecționeze pachetele pe baza unui cod unic de identificare și a unui senzor de greutate.

Specificații și cerințe:

- Detectia și identificarea pachetelor:**
 - Fiecare pachet are un **cod binar de 4 biți** care reprezintă destinația (ex: 1011 pentru zona B).
 - Un **senzor de greutate** detectează pachetele care depășesc limita maximă admisă (ex: 5 kg).
- Sortare și direcționare:**
 - Un **circuit combinational** decodează codul pachetului și activează motorul benzii transportoare pentru a-l trimite spre zona corectă.
 - Dacă pachetul este prea greu, acesta este trimis automat spre zona de inspecție.
 - Dacă codul pachetului nu este recunoscut, acesta este trimis într-o zonă de eroare.
- Automatizare și procesare secvențială:**
 - Sistemul funcționează în mod **secvențial**, procesând un singur pachet la un moment dat.
 - Utilizarea unui **automat de stare finit (FSM)** pentru gestionarea etapelor procesului:
 - Detectare pachet.
 - Citire cod și greutate.
 - Decizie și activare direcționare.
 - Verificare finală și procesare următor pachet.
- Interfață de monitorizare:**
 - Un afișaj cu **LED-uri indicatoare** semnalează starea sistemului (procesare, eroare, zona de destinație activă).
 - Un buton de **reset manual** permite repornirea sistemului în caz de blocaj.

DESCRIEREA TEMEI – Semafor Intelligent cu Priorizare a Vehiculelor de Urgență

CERINȚĂ

Se va proiecta un sistem de control al unui semafor intelligent, capabil să regleze traficul în funcție de fluxul de mașini și să acorde prioritate vehiculelor de urgență (ex: ambulanță, pompieri, poliție). Sistemul trebuie să folosească circuite logice combinaționale și secvențiale pentru gestionarea schimbării culorilor semaforului.

Specificații și cerințe:

- Funcționarea standard a semaforului:**
 - Sistemul controlează **un semafor cu 3 culori** (verde, galben, roșu).
 - Comutarea între stări se face automat în funcție de un **cronometru intern**.
 - Timpii standard de funcționare sunt:
 - Verde:** 10 secunde
 - Galben:** 3 secunde
 - Roșu:** 10 secunde
- Detectia fluxului de trafic:**
 - Senzori de **presiune** sau **infraroșu** detectează prezența vehiculelor la semafor.
 - Dacă nu există mașini, semaforul rămâne pe roșu pentru economisirea energiei.
 - Dacă fluxul de trafic depășește un prag critic, semaforul prelungește timpul de verde.
- Priorizarea vehiculelor de urgență:**
 - Un **semnal radio** emis de un vehicul de urgență activează un **circuit combinational** care schimbă semaforul pe verde instantaneu.
 - După trecerea vehiculului de urgență, semaforul revine la funcționarea normală.
- Automat de stare finit (FSM) pentru controlul semaforului:**
 - FSM gestionează **stările semaforului** în funcție de timp, trafic și vehicule de urgență.
 - Stările FSM:
 - Stare normală** (trecere între verde-galben-roșu)
 - Timp ajustabil** (prelungirea semnalului verde în funcție de trafic)
 - Prioritate vehicule de urgență** (comutare forțată pe verde)

DESCRIEREA TEMEI – Sistem Automat de Irigare Intelligentă pentru Seră

CERINȚĂ

Se va proiecta un sistem automat de irigare intelligentă, care ajustează cantitatea de apă furnizată plantelor în funcție de umiditatea solului și de condițiile de mediu. Sistemul trebuie să utilizeze circuite logice combinaționale și secvențiale pentru luarea deciziilor privind activarea sau dezactivarea pompei de apă.

Specificații și cerințe:

- 1. Monitorizarea condițiilor din seră:**
 - Se utilizează **un senzor de umiditate a solului** pentru a detecta nivelul apei din pământ.
 - Un **senzor de temperatură și lumină** influențează decizia de irigare (ex: nu se udă în plină zi dacă temperatura este ridicată).
- 2. Controlul automat al irigării:**
 - Sistemul folosește **un circuit combinational** pentru a decide dacă trebuie activată irigarea, pe baza următoarelor condiții:
 1. **Umiditate scăzută + temperatură ridicată = activare irigare**
 2. **Umiditate optimă = oprire irigare**
 3. **Ploaie detectată (prin senzor de umiditate atmosferică) = oprire irigare**
 - Irigarea este efectuată pentru un timp determinat (ex: 5 minute) sau până când solul atinge umiditatea optimă.
- 3. Automat de stare finit (FSM) pentru gestionarea procesului de irigare:**
 - FSM va avea următoarele stări:
 1. **Așteptare** (monitorizare senzori)
 2. **Activare pompa** (dacă este necesară irigarea)
 3. **Irigare în curs** (controlul duratei)
 4. **Oprise automată** (când umiditatea solului este suficientă)
- 4. Interfață de monitorizare:**
 - Un afișaj cu **LED-uri** indică starea sistemului (irigare activă, oprire, eroare).
 - Se poate adăuga un **buzzer** care alertează dacă senzorii detectează o problemă (ex: lipsă apă în rezervor).

DESCRIEREA TEMEI – Sistem de Monitorizare și Avertizare pentru Cutremure

CERINȚĂ

Se va proiecta un sistem automat de monitorizare și avertizare pentru cutremure, capabil să detecteze vibrațiile solului și să declanșeze alarme în cazul unui seism puternic. Sistemul va utiliza circuite logice combinaționale și secvențiale pentru interpretarea datelor de la senzori și luarea deciziilor.

Specificații și cerințe:

- Detectia vibrațiilor:**
 - Sistemul va folosi **un accelerometru** pentru a măsura mișcările solului.
 - Se definesc **trei niveluri de alertă** pe baza amplitudinii semnalului:
 - Nivel scăzut** – vibrații normale, fără alarmă.
 - Nivel mediu** – vibrații suspecte, se activează un semnal de avertizare vizuală.
 - Nivel ridicat** – cutremur detectat, se activează alarma sonoră și vizuală.
- Procesarea datelor și luarea deciziilor:**
 - Un circuit combinational** interpretează datele primite de la accelerometru.
 - Un circuit secvențial** gestionează timpul și persistența vibrațiilor pentru a evita alarme false.
 - Dacă nivelul de vibrație ridicat este detectat timp de **3 secunde consecutive**, se activează alarma generală.
- Automat de stare finit (FSM) pentru gestionarea alertelor:**
 - FSM va avea următoarele stări:
 - Monitorizare** (citirea constantă a senzorului)
 - Avertizare** (dacă vibrațiile sunt moderate, activare LED de avertizare)
 - Alarmă activă** (dacă vibrațiile persistă, activare sirenă și afișaj „Cutremur detectat!”)
 - Reset automat** (dacă vibrațiile scad sub pragul critic)
- Sistem de notificare:**
 - Un buzzer** și **un LED roșu intermitent** indică o alertă gravă.
 - Un afișaj LCD sau un set de LED-uri indică nivelul actual de vibrație.

DESCRIEREA TEMEI – Joc Electronic „Simon Says” cu Logica Programată

CERINȚĂ

Se va proiecta un joc electronic de tip „**Simon Says**”, în care utilizatorul trebuie să repete o secvență de lumini și sunete generată aleatoriu de sistem. Proiectul trebuie să utilizeze **circuite logice combinaționale și secvențiale** pentru generarea secvențelor și verificarea răspunsurilor utilizatorului.

Specificații și cerințe:

- 1. Generarea secvenței de joc:**
 - Sistemul generează aleatoriu o **secvență de LED-uri colorate** (ex: roșu-verde-albastru-galben).
 - Secvența crește în dificultate pe măsură ce jucătorul răspunde corect.
 - **Un generator de numere pseudoaleatorii** stabilește ordinea culorilor.
- 2. Interacțiunea utilizatorului:**
 - Jucătorul trebuie să repete secvența apăsând **butonul corespunzător fiecărei culori**.
 - Fiecare apăsare este verificată de un **circuit secvențial** pentru a corespunde cu secvența corectă.
 - Dacă utilizatorul greșește, se activează un **buzzer** și jocul se resetează.
- 3. Automat de stare finit (FSM) pentru gestionarea jocului:**
 - FSM controlează **stările jocului**:
 1. **Stare inițială** – așteaptă startul jocului.
 2. **Generare secvență** – adaugă o nouă culoare la secvență.
 3. **Așteptare input utilizator** – verifică dacă jucătorul repetă corect secvența.
 4. **Verificare răspuns** – compară input-ul utilizatorului cu secvența memorată.
 5. **Victorie** – dacă secvența este completată corect, adaugă o nouă culoare și continuă jocul.
 6. **Game Over** – dacă utilizatorul greșește, jocul se resetează.
- 4. Indicatori vizuali și sonori:**
 - **LED-uri colorate** semnalizează secvența generată.
 - **Buzzer-ul** emite un sunet specific fiecărei culori.
 - **Un afișaj cu 7 segmente sau LCD** indică scorul curent (numărul de runde complete).

DESCRIEREA TEMEI – Sistem Automat de Sortare a Monedelor

CERINȚĂ

Se va proiecta un sistem automatizat care poate identifica și sorta monedele în funcție de dimensiune și greutate. Sistemul trebuie să utilizeze **circuite logice combinaționale și secvențiale** pentru a determina valoarea monedei și a direcționa fiecare monedă către compartimentul corespunzător.

Specificații și cerințe:

1. Detectarea și identificarea monedelor:

- Sistemul utilizează un **senzor optic sau un senzor de lumină** pentru a măsura diametrul monedei.
- Un **senzor de greutate** determină masa monedei pentru validare suplimentară.
- Pe baza acestor măsurători, un **circuit combinational** determină tipul monedei (ex: 5 bani, 10 bani, 50 bani).

2. Sortarea și direcționarea monedelor:

- Un **circuit secvențial** gestionează direcționarea monedelor către compartimentele corecte printr-un sistem de clapete controlate electronic.
- Dacă moneda nu este recunoscută (dimensiune sau greutate incorectă), este direcționată într-un compartiment de eroare.

3. Automat de stare finit (FSM) pentru gestionarea procesului de sortare:

- FSM controlează stările sistemului:
 1. **Așteptare** – sistemul este pregătit să primească o monedă.
 2. **Măsurare** – senzorii determină dimensiunea și greutatea monedei.
 3. **Comparare** – circuitul decide valoarea monedei pe baza măsurătorilor.
 4. **Directionare** – activarea mecanismului pentru trimiterea monedei în compartimentul corect.
 5. **Reset** – sistemul revine la starea inițială pentru următoarea monedă.

4. Indicatori vizuali și sonori:

- Un **afișaj cu 7 segmente sau LCD** poate indica valoarea ultimei monede detectate.
- Un **LED roșu și un buzzer** se activează dacă moneda este invalidă sau necunoscută.

DESCRIEREA TEMEI – Sistem de Control al Energiei într-o Clădire Inteligentă

CERINȚĂ

Se va proiecta un sistem automatizat care monitorizează consumul de energie electrică într-o clădire și optimizează utilizarea acesteia prin oprirea sau reglarea dispozitivelor electrice în funcție de necesități. Sistemul trebuie să utilizeze **circuite logice combinaționale și secvențiale** pentru luarea deciziilor privind gestionarea consumului.

Specificații și cerințe:

1. Monitorizarea consumului de energie:

- Sistemul utilizează **senzori de curent și tensiune** pentru a măsura consumul fiecărei zone a clădirii.
- Datele sunt procesate de un **circuit combinational** care compară consumul curent cu limitele prestabilite.
- Dacă o zonă depășește pragul de consum permis, sistemul ia măsuri pentru reducerea acestuia.

2. Gestionarea automată a dispozitivelor:

- Dacă consumul este prea mare, sistemul poate **opri temporar anumite aparete electrice neesențiale** (ex: lumini, aer condiționat, prize inteligente).
- Dacă detectează **prezență umană** (prin senzori PIR sau de mișcare), reaprinde luminile automat.
- Se poate activa un **mod economic**, care reduce treptat consumul prin optimizarea funcționării dispozitivelor.

3. Automat de stare finit (FSM) pentru gestionarea consumului:

- FSM controlează **stările sistemului**:
 1. **Monitorizare normală** – consumul este în limite normale, toate dispozitivele funcționează.
 2. **Atenționare** – consumul este aproape de limită, utilizatorul este avertizat prin LED sau buzzer.
 3. **Reducere consum** – anumite dispozitive sunt opriate sau trecute în mod economic.
 4. **Restaurare** – dacă nivelul de consum revine la normal, dispozitivele sunt reactivate treptat.

4. Indicatori vizuali și sonori:

- **Un afișaj LCD sau LED-uri** arată consumul curent de energie și starea sistemului.
- **Un buzzer** se activează dacă sistemul trebuie să reducă drastic consumul.
- Se poate adăuga o interfață pentru **control manual** al priorităților (ex: utilizatorul poate decide ce dispozitive să rămână active).

DESCRIEREA TEMEI – Sistem de Prelucrare a Codurilor Morse

CERINȚĂ

Să se implementeze din punct de vedere software un sistem de prelucrare a codurilor Morse.

Specificații și cerințe:

1. Generarea și prelucrarea semnalelor

- Semnalele se vor genera folosind 2 pini ai unui port setat ca intrare.
- Cele 2 intrări vor fi asociate structurilor de cod aferente: punct (PINX.i) și linie (PINX.j).
- Separarea caracterelor ASCII transmise prin cuvintele de cod ale codului MORSE (punct și linie) se va realiza într-un timp scurt predefinit în care utilizatorul nu a apăsat pe niciun buton (PIN). Sistemul va rămâne în așteptare până la următoarea intrare asociată oricărui buton, marcând așteptarea cu un LED aprins (la următorul input se va stinge LED-ul indicator).

2. Exemplu

Fie timpul de așteptare o secundă.

- Utilizatorul emite caracterul A (punct linie) -> sistemul stochează intrările într-un buffer -> sistemul contorizează timpul în care nu a mai primit un input -> sistemul ajunge la o secundă -> Se vor prelucra datele din buffer căutând caracterul asociat codului Morse -> se aprinde LED-ul indicator -> se așteaptă următorul input etc.
- Pe scurt, dacă trece mai mult de o secundă de la ultimul buton apăsat (adică de la ultimul cuvânt de cod recepționat), sistemul va prelucra cadrul curent de informații, succesiv întrând într-o stare de așteptare a unui nou set de cuvinte de cod (input-uri).
- Dacă la un moment dat, fie la început, final sau în timpul transmisiei de date, se recepționează semnalul SOS, sistemul se va bloca aprinzând toate ledurile la un interval de 0.5 secunde, timp de 10 secunde (se va folosi același port de ieșire configurat anterior), ștergând toată transmisia anterioară semnalului de SOS și așteptând una nouă.

3. Sfaturi pentru proiectare

- Pentru citirea intrării folosiți întreruperi externe, evitați polling.
- Utilizarea unui hash simplu pentru căutarea codurilor ASCII într-un buffer

predefinit.

- Înglobarea modulelor software (a procedurilor folosite) pentru liniaritate.
- Dezvoltarea sistemului pe stări (Proces Secvențial - Finite State Machine).
- Utilizarea flag-urilor software atât statice în funcții, cât și globale, în funcție de caz.
- Evitarea dezvoltării codului în ISR dacă același cod poate fi scris în bucla principală.