

Proiectarea sistemelor numerice

Implementare în Active-HDL, în cod VHDL
cu descriere top-level structurală
(fără implementare pe plăcuță FPGA)



Profesor îndrumător:
As. Ing. Pop Diana
Student:
Pop Raul-George

Cuprins

1. Specificațiile proiectului	3
1.1 Sarcina de lucru.....	3
1.2 Studiu teoretic	3
1.3 Algoritmul de funcționare	5
2. Proiectare.....	6
2.1 Schema bloc	6
2.1.1 Schema.....	6
2.1.2 Listă de intrări/ieșiri	7
2.2 Descompunerea în Unitate de control și Unitate de execuție.....	8
2.2.1 Lista completă a resurselor	9
2.3 Descrierea detaliată a resurselor	9
2.4 Organigrama Unității de comandă	17
2.5 Schema de detaliu a Unității de execuție	18
2.6 Sinteza Unității de comandă	20
3. Justificarea soluției alese	21
3.1 Alegerea resurselor	21
3.2 Decizii de design.....	22
4. Manual de utilizare și întreținere	23
4.1 Utilizarea aparatului.....	23
4.2 Semnificația butoanelor	23
4.3 Administrarea aparatului și alte atenționări	24
5. Posibilități de dezvoltare	24
Bibliografie	25

1. Specificațiile proiectului

1.1 Sarcina de lucru

B5) Să se proiecteze un automat pentru un ceas cu radio, programabil. Ceasul poate fi resetat, fixat la o oră dorită, pornit, oprit. Ceasul poate fi programat să pornească și să oprească o sonerie sau să pornească și să oprească radioul.

1.2 Studiu teoretic

Având în vedere că trebuie să implementăm un ceas digital, vom avea nevoie obligatoriu de 4 afișaje cu LED-uri cu 7 segmente, pentru a reprezenta ora.

Afișajul cu șapte segmente este o modalitate de a reprezenta numere în sisteme electronice. Este format din șapte segmente care pot fi activate sau dezactivate individual, fiecare segment având forma unei linii scurte. Fiecare dintre segmentele care formează afișajul sunt numite a, b, c, d, e, f și g sunt asamblate astfel încât să permită a activa fiecare segment separat, astfel obținând orice cifră (*Fig. 1* și *Fig. 2*).

Display-urile cu șapte segmente sunt de două tipuri: *anod comun* și *catod comun*.

Pentru tipul *anod comun*, tot circuitul de tip anod sau segmente sunt conectate intern la un pin comun pentru a fi conectat la potențialul pozitiv (nivel "1"). Aprinderea individuală a fiecărui segment este realizată prin aplicarea unui potențial negativ (nivel "0") pentru pinul corespunzător printr-o rezistență care limitează fluxul de curent.

La tipul *catod comun*, toți catodii de LED-uri sau segmente sunt unite intern la un catod comun pentru a fi conectați la potențialul negativ (nivel "0"). Aprinderea individuală a fiecărui segment se face prin aplicarea unui potențial pozitiv (nivel "1") care corespunde printr-o rezistență de limitare a fluxului de curent.

Caracter	a	b	c	d	e	f	g
0	0	0	0	0	0	0	1
1	1	0	0	1	1	1	1
2	0	0	1	0	0	1	0
3	0	0	0	0	1	1	0
4	1	0	0	1	1	0	0
5	0	1	0	0	1	0	0
6	0	1	0	0	0	0	0
7	0	0	0	1	1	1	1
8	0	0	0	0	0	0	0
9	0	0	0	0	1	0	0

Fig. 1 Valorile segmentelor și caracterul corespunzător afișat

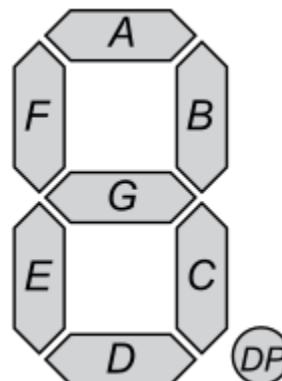


Fig.2 Codificarea celor 7 segmente

Un **bistabil** este un circuit care are două stări stabile și poate fi utilizat pentru a stoca informații de stare. Circuitul poate fi realizat pentru a schimba starea prin semnale aplicate la una sau mai multe intrări de control și va avea una sau două ieșiri. Este elementul de stocare de bază în logica secvențială.

Bistabilele flip-flop sunt utilizate ca elemente de stocare a datelor. Un flip-flop este un dispozitiv care stochează un singur bit (cifră binară) de date. Acestea funcționează în raport cu un semnal de tact (clock).

Numărătoarele sunt circuitele logice secvențiale (CLS) care contorizează numărul de impulsuri aplicate pe intrarea de tact. Acestea vor avea rolul de a urmări timpul. Se implementează cu bistabile JK sau T ca suport de memorare a stării binare reprezentată pe mai mulți biți. Cu n bistabile se pot reprezenta maxim 2^n numere binare pe n biți. Fiecare stare are asociată un număr. Secvența de numere prin care trece starea număratorului pe măsură ce înregistrează impulsuri reprezintă codul binar în care numără număratorului.

1.3 Algoritm de funcționare

Având în vedere că trebuie să implementăm un ceas digital ne vom folosi de afișaje pe 7 segmente pentru a reprezenta ora și de diferite numărătoare pentru a urmări ora. Pentru a implementa mecanismul efectiv al ceasului, vom avea nevoie de un semnal de tact care va reprezenta o secundă. Cu ajutorul a două numărătoare modulo 60, vom număra de la "00" la "59" atât secunde cât și minute astfel: în momentul în care numărătorul responsabil cu numărarea secundelor ajunge la valoarea 59, numărătorul responsabil cu numărarea minutelor va fi incrementat iar când și acesta va ajunge la valoarea 59, un alt numărător, modulo 24, va fi incrementat (numărător responsabil cu numărarea orelor).

Ceasul are opțiunea de a fi resetat, deci vom avea nevoie de un buton de reset asincron care va fi conectat la toate numărătoarele menționate anterior.

Ceasul are opțiunea de a fi fixat la o oră dorită, deci vom avea nevoie de 2 butoane pentru a incrementa manual numărătoarele ceasului, responsabile cu numărarea orelor respectiv minutelor, până la ora dorită. Pentru a semnaliza faptul că setăm ceasul, vom avea nevoie de un buton care să oprească funcționarea normală a acestuia și care să ne permită incrementarea manuală a numărătoarelor.

Ceasul are opțiunea de a porni/opri o sonerie la o oră dată, deci vom avea nevoie de un buton de setare respectiv de oprire a alarmei.

Aparatul are opțiunea de a fi pornit/oprit, deci vom avea nevoie de un buton de pornire/oprire.

Ceasul are opțiunea de a porni/opri radioul, deci vom avea nevoie de un buton de pornire/oprire pentru radio.

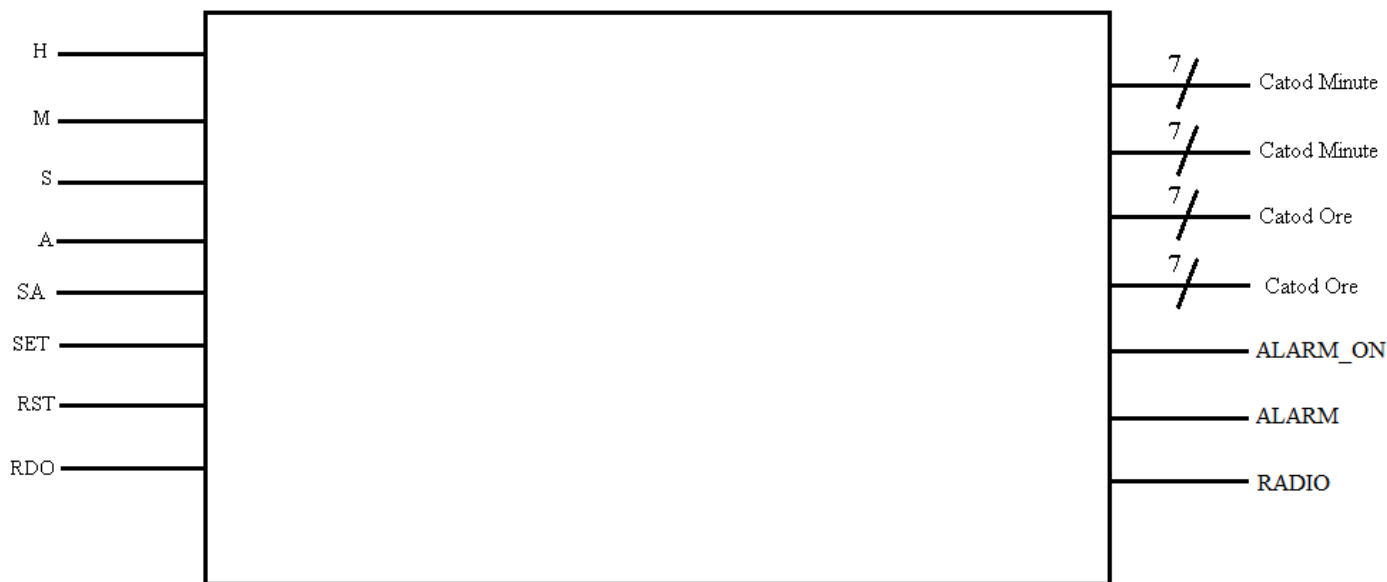
În momentul pornirii aparatului, numărătoarele încep să se incrementeze. Utilizatorul are opțiunea de a seta ceasul la o oră dorită. În momentul apăsării butonului de *SET*, numărătoarele

trec pe modul de incrementare manuală și vor permite astfel utilizatorului să le incrementeze până la o oră dorită. Prin apăsarea butonului de *SET*, ceasul va reveni la modul standard și va continua în mod automat numărarea. În cazul în care utilizatorul va dori setarea unei alarme, acesta va fi nevoit să apese butonul *ALARM* care va comuta informația afișată pe afișaje de la numărătoarele ceasului la alte numărătoare dedicate setării alarmei, numărătoare care au incrementare manuală și care vor păstra valoarea introdusă de utilizator pentru a fi ulterior comparată. În acest fel, utilizatorul va putea seta alarma fără a pierde informații și a fi nevoit să seteze iar ora.

2.Proiectare

2.1 Schema bloc

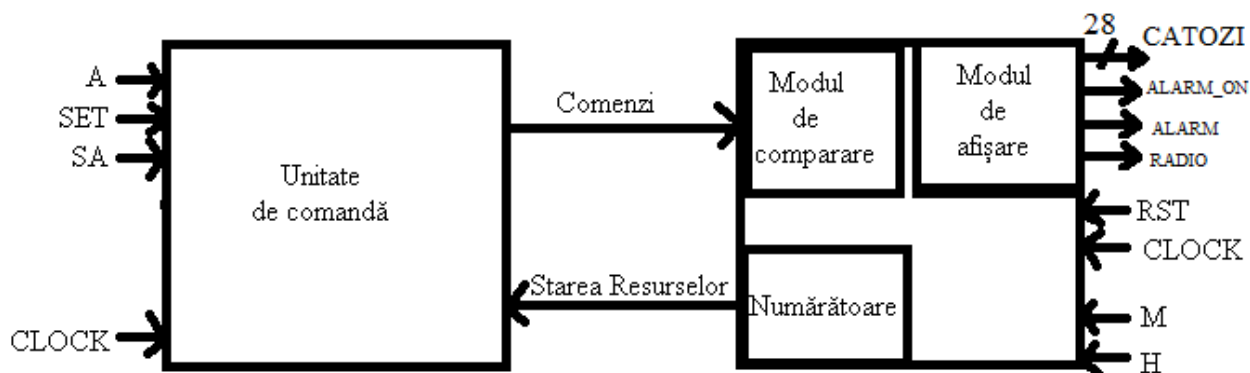
2.1.1 Schema



2.1.2 Listă de intrări/ieșiri

- H – *HOUR* – va reprezenta o intrare simplă pe 1 bit folosită la incrementarea orei pentru setarea ceasului sau a alarmei.
- M – *MINUTES* – va reprezenta o intrare simplă pe 1 bit folosită la incrementarea minutelor pentru setarea ceasului sau a alarmei.
- S – *START/STOP*- va reprezenta o intrare simplă pe 1 bit și va avea rolul de a porni/opri aparatul.
- A – *ALARM* – va reprezenta o intrare simplă pe 1 bit și va avea rolul de a oferi utilizatorului opțiunea de a seta alarma.
- SA – *STOP ALARM* - va reprezenta o intrare simplă pe 1 bit și va avea rolul de a opri alarma.
- SET – *SET* – va reprezenta o intrare simplă pe 1 bit și va avea rolul de a oferi utilizatorului opțiunea de a seta ceasul.
- RST – *RESET* – va reprezenta o intrare simplă pe 1 bit și va avea rolul de a reseta ceasul.
- RDO – *RADIO* – va reprezenta o intrare simplă pe 1 bit și va avea rolul de a porni/opri radioul.
- Catozii – vor reprezenta magistrale pe 7 biți care vor avea rolul de a activa diferite segmente dintr-un afișaj.
- ALARM_ON – va reprezenta o ieșire simplă pe 1 bit și va avea rolul de a semnala dacă alarma a fost sau nu setată.
- ALARM – va reprezenta o ieșire simplă pe 1 bit și va avea rolul de a semnala declanșarea alarmei.
- RADIO – va reprezenta o ieșire simplă pe 1 bit și va avea rolul de a semnala pornirea/oprirea radioului.

2.2 Descompunerea în Unitate de control și Unitate de execuție



Sistemele numerice complexe pot fi văzute ca fiind compuse dintr-o *unitate de comandă* – UC (sistemul logic secvențial care implementează algoritmul propriu-zis) și o *unitate de execuție* – UE (colecție de resurse care realizează operațiunile elementare).

Unitatea de comandă este cea care controlează în mod individual fiecare resursă din unitatea de execuție, furnizându-i acesteia din urmă semnalele de comandă pentru toate resursele sale în configurația potrivită și la momentul potrivit, astfel încât algoritmul să fie dus la bun sfârșit.

Pentru a lua decizii, UC are nevoie să citească atât variabilele de intrare ale sistemului, cât și starea resurselor din UE. Atât intrările cât și ieșirile sistemului pot fi citite/generate atât în/din UC cât și în/din UE.

În cazul de față, UE va fi alcătuită dintr-un modul de numărare, un modul de comparare și de un modul de afișare. UC va fi un circuit logic secvențial care va comanda modulul de afișare al ceasului și va lua decizii atât în funcție de butoanele asincrone ale sistemului (ALARM, STOPALARM, SET) cât și de numărătoarele din UE. Întreg sistemul poate fi considerat sincron: UC și UE funcționează pe același tact, însă pentru a asigura o bună sincronizare a componentelor cu afisajele, vom avea nevoie de un divizor de frecvență.

2.2.1 Lista completă a resurselor

- Numărător modulo 60
- Numărător modulo 24
- Decodificator pentru afișaje pe 7 segmente
- Comparator numeric pe 4 biți

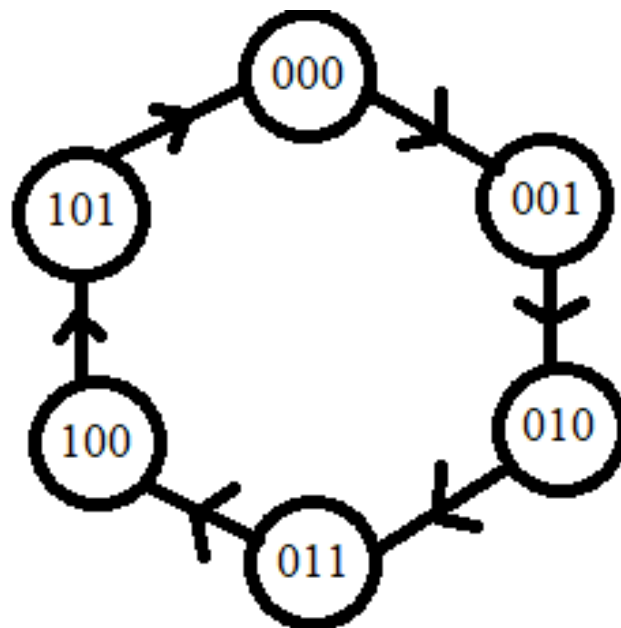
2.3 Descrierea detaliată a resurselor

Numărător modulo 60

Pentru a putea afișa ora pe afișaje cu 7 segmente, vom realiza numărătorul modulo 60 din 2 numărătoare: modulo 6 (pentru cifra zecilor) respectiv modulo 10 (pentru cifra unităților).

- Numărătorul modulo 6

Graful de tranziții al numărătorului este:



Tabelul de excitație este:

Q_2^t	Q_1^t	Q_0^t	Q_2^{t+1}	Q_1^{t+1}	Q_0^{t+1}	J_2	K_2	J_1	K_1	J_0	K_0
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	0	1	1	0	X	X	0	1	X
0	1	1	1	0	0	1	X	X	1	X	1
1	0	0	1	0	1	X	0	0	X	1	X
1	0	1	0	0	0	X	1	0	X	X	1

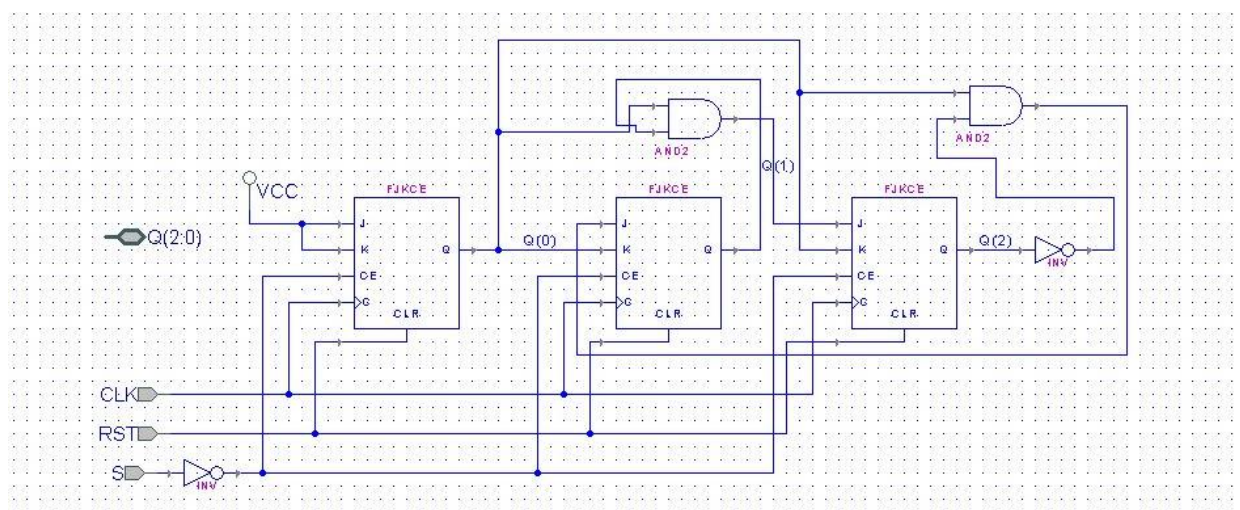
Realizând diagramele Karnaugh, minimizăm funcțiile rezultate și obținem:

$$J_2 = Q_1^t \bullet Q_0^t \quad K_2 = Q_0^t$$

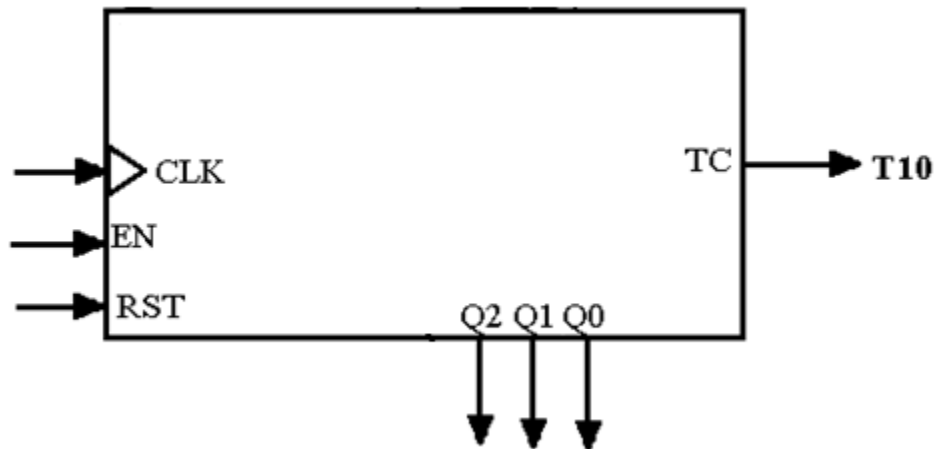
$$J_1 = (\neg Q_2^t) \bullet Q_0^t \quad K_1 = Q_0^t$$

$$J_0 = 1 \quad K_0 = 1$$

Schema logică a numărătorului este:

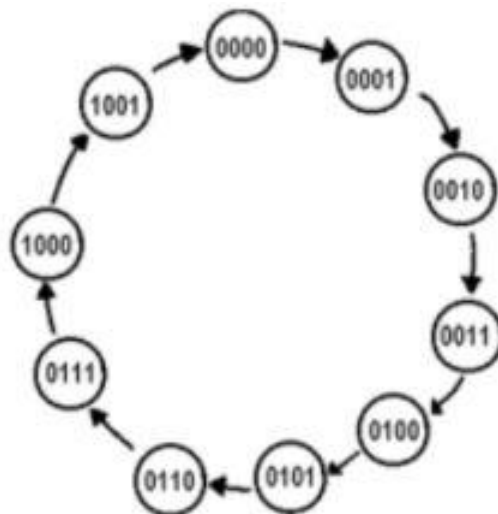


Schema bloc a numărătorului este:



- Numărător modulo 10 (zecimal)

Graful de tranziții al numărătorului este:



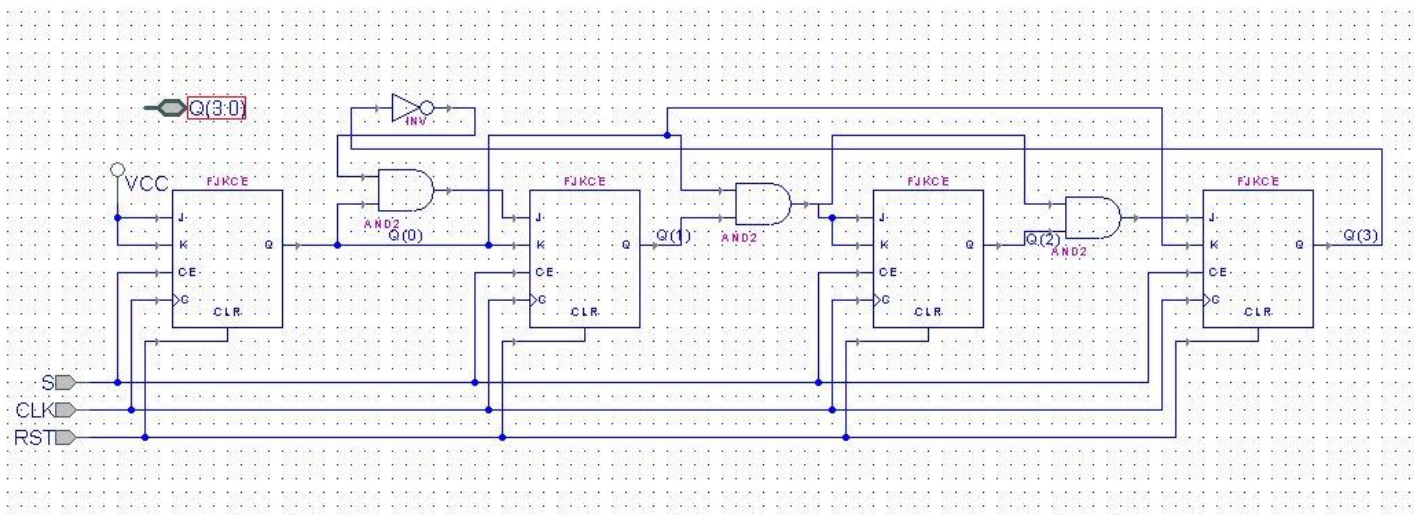
Tabelul de excitație este:

Q ₃	Q ₂	Q ₁	Q ₀	Q ₃	Q ₂	Q ₁	Q ₀	J ₃	K ₃	J ₂	K ₂	J ₁	K ₁	J ₀	K ₀
0	0	0	0	0	0	0	1	0	X	0	X	0	X	1	X
0	0	0	1	0	0	1	0	0	X	0	X	1	X	X	1
0	0	1	0	0	0	1	1	0	X	0	X	X	0	1	X
0	0	1	1	0	1	0	0	0	X	1	X	X	1	X	1
0	1	0	0	0	1	0	1	0	X	X	0	0	X	1	X
0	1	0	1	0	1	1	0	0	X	X	0	1	X	X	1
0	1	1	0	0	1	1	1	0	X	X	0	X	0	1	X
0	1	1	1	1	0	0	0	1	X	X	1	X	1	X	1
1	0	0	0	1	0	0	1	X	0	0	X	0	X	1	X
1	0	0	1	0	0	0	0	X	1	0	X	0	X	X	1

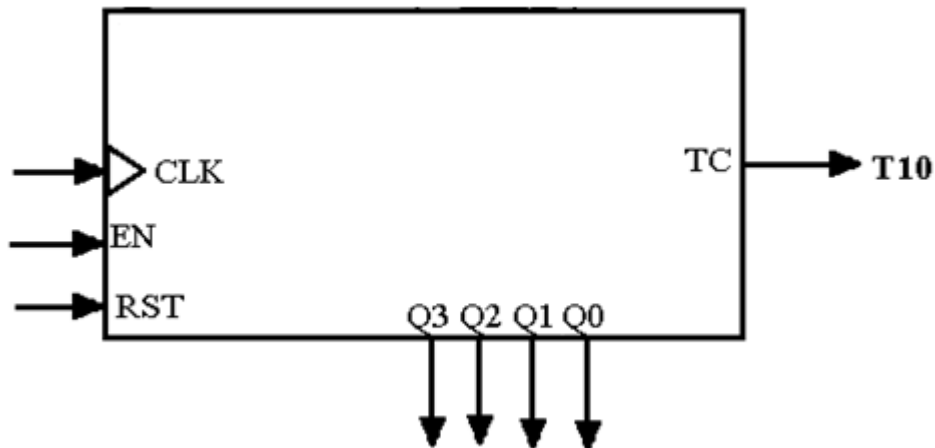
Realizând diagramele Karnaugh, minimizăm funcțiile rezultate și obținem:

$$\begin{aligned}
 J_3 &= Q_2 \bullet Q_1 \bullet Q_0 & K_3 &= Q_0 & J_2 &= Q_1 \bullet Q_0 & K_2 &= Q_1 \bullet Q_0 \\
 J_1 &= (\neg Q_3) \bullet Q_0 & K_1 &= Q_0 & J_0 &= 1 & K_0 &= 1
 \end{aligned}$$

Schema logică a numărătorului este:



Schema bloc a numărătorului este:

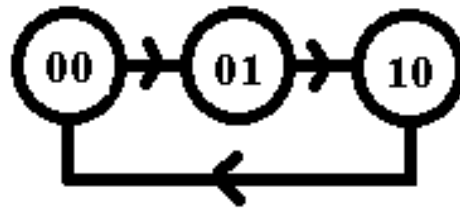


Numărător modulo 24

Pentru a putea afișa ora pe afișaje cu 7 segmente, vom realiza numărătorul modulo 24 din 2 numărătoare: modulo 3 (pentru cifra zecilor) respectiv modulo 10 (pentru cifra unităților).

- Numărătorul modulo 3

Graful de tranziții al numărătorului este:



Tabelul de excitație este:

Q_1^t	Q_0^t	Q_1^{t+1}	Q_0^{t+1}	J_1	K_1	J_0	K_0
0	0	0	1	0	X	1	X
0	1	1	0	1	X	X	1
1	0	0	0	X	1	0	X

Realizând diagramele Karnaugh, minimizăm funcțiile rezultate și obținem:

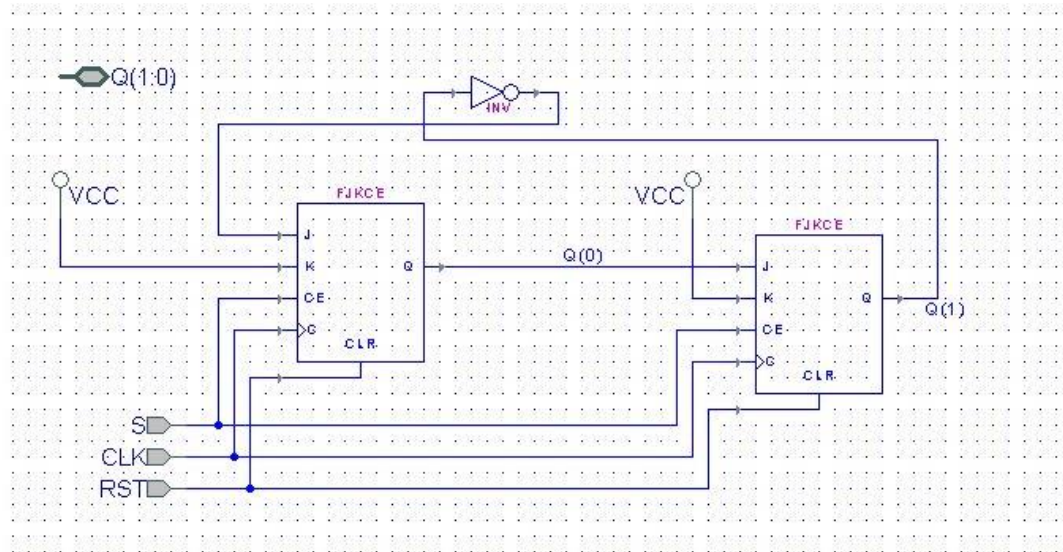
$$J_1 = Q_0^t$$

$$K_1 = 1$$

$$J_0 = \neg Q_1^t$$

$$K_0 = 1$$

Schema logică a numărătorului este:



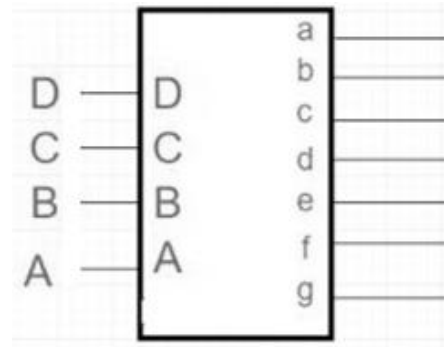
Decodificator pentru afișaje 7 segmente

Pentru a realiza afișarea pe 7 segmente avem nevoie mai întâi de un decodificator care să transforme un număr binar într-un cod pe 7 segmente.

Tabelul de adevăr este:

A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1

Schema bloc a acestui decodificator este:



Realizând diagramele Karnaugh, minimizăm funcțiile rezultate și obținem:

$$a = ((\neg B) \bullet (\neg D)) + C + (B \bullet D) + A$$

$$b = (\neg B) + ((\neg C) \bullet (\neg D)) + (C \bullet D)$$

$$c = (\neg C) + D + B$$

$$d = ((\neg B) \bullet (\neg D)) + ((\neg B) \bullet C) + (C \bullet (\neg D)) + ((B \bullet ((\neg C) \bullet D))) + A$$

$$e = ((\neg B) \bullet (\neg D)) + (C \bullet (\neg D))$$

$$f = ((\neg C) \bullet (\neg D)) + (B \bullet (\neg C)) + (B \bullet (\neg D)) + A$$

$$g = ((\neg B) \bullet C) + (B \bullet (\neg C)) + (B \bullet (\neg D)) + A$$

Rezultatul numărătoarelor vor intra fiecare într-un astfel de decodificator, apoi pe rând, vor intra pe catodii afișoarelor pe 7 segmente.

Comparatoare numerice

Pentru a compara datele introduse de utilizator cu datele curente vom avea nevoie de comparatoare numerice care să verifice acest lucru.

Tabelul de adevăr al unui comparator pe 1 bit este:

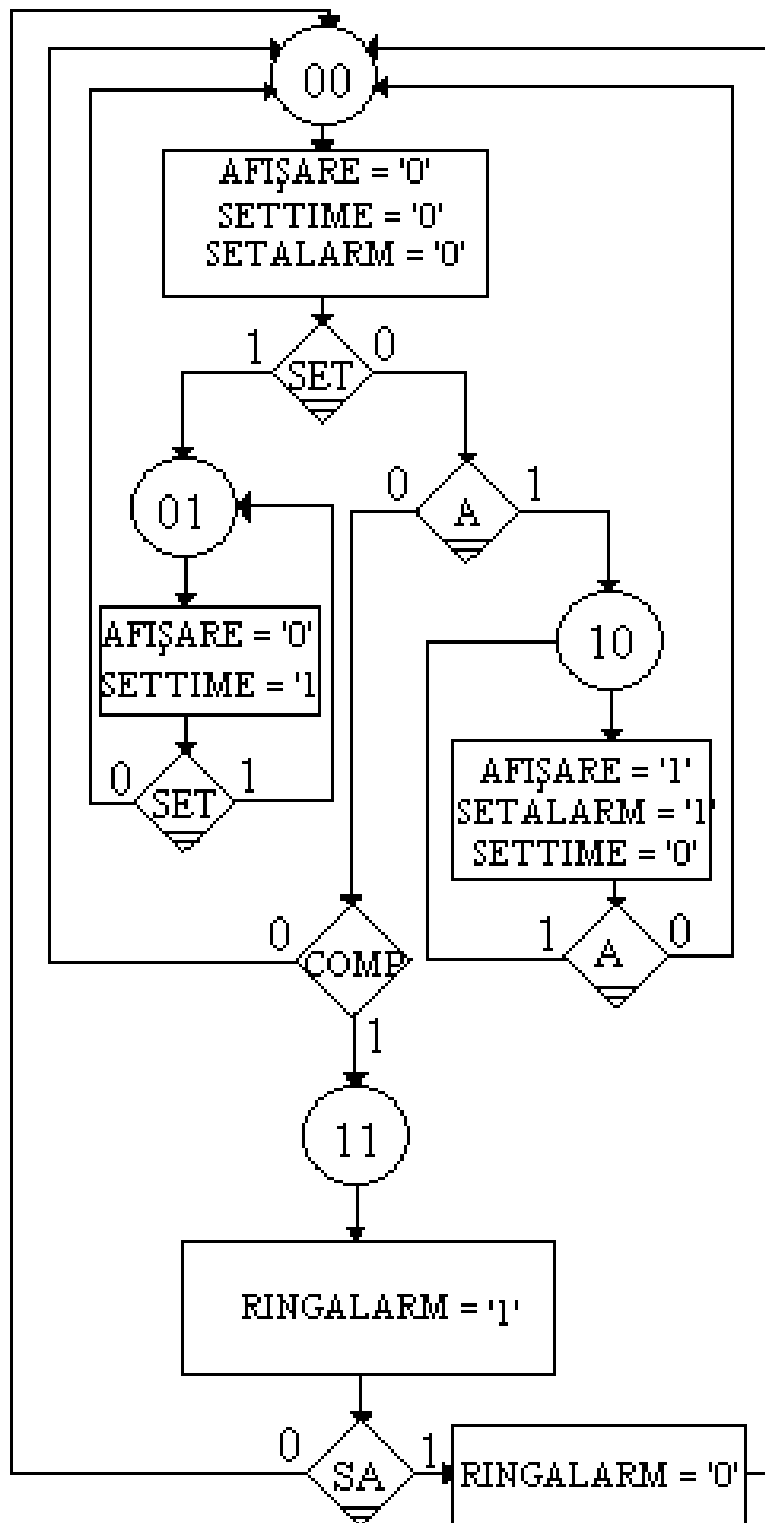
B	A	$A > B$	$A = B$	$A < B$
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

Realizând diagramele Karnaugh, minimizăm funcțiile rezultate și obținem funcția de egalitate:

$$F_{=} = A \odot B$$

Deoarece în cadrul proiectului vom avea nevoie de comparatoare numerice pe mai mulți biți, vom cascada mai multe comparatoare numerice pe 1 bit.

2.4 Organigrama Unității de comandă



Codificarea stărilor:

00 -> Stare de așteptare (Counting);

01 -> Stare de setare a ceasului;

10 -> Stare de setare a alarmei;

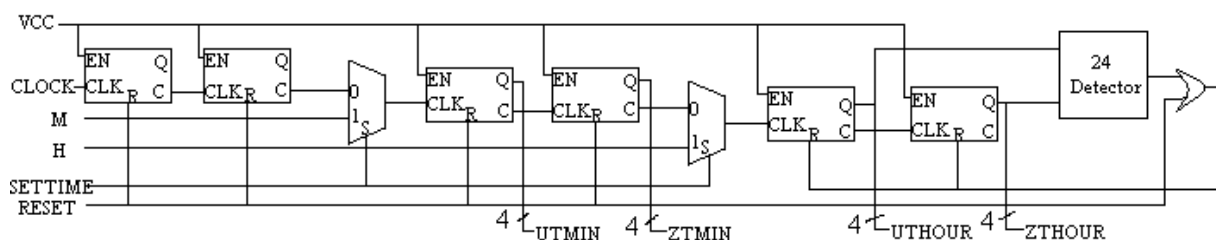
11 -> Stare de declanșare a soneriei.

2.5 Schema de detaliu a Unității de execuție

Unitatea de execuție este alcătuită din mai multe module, fiecare având un rol diferit.

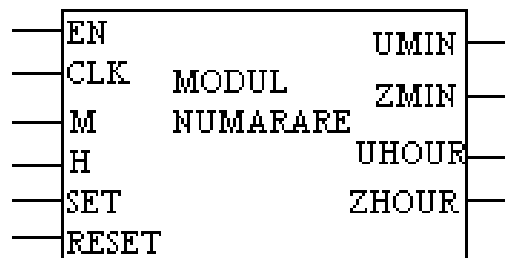
1.Modulul de numărare: Este alcătuit dintr-o serie de numărătoare cu rolul de a număra secunde, minutele și orele.

Schema logică a modulului este:



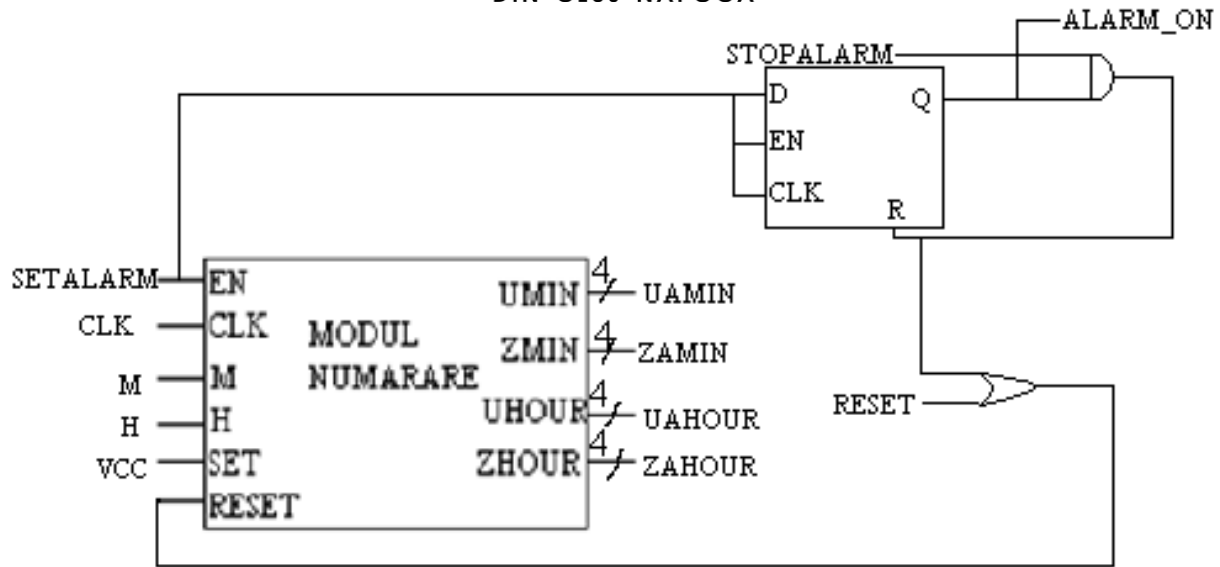
Schema bloc a modulului este:

Pentru a memora datele pentru alarma vom folosi același modul însă cu intrări diferite și la care vom adăuga un bistabil D care va avea rolul de a memora faptul că am setat alarma. Astfel, pentru modulul alarmei vom avea intrările:

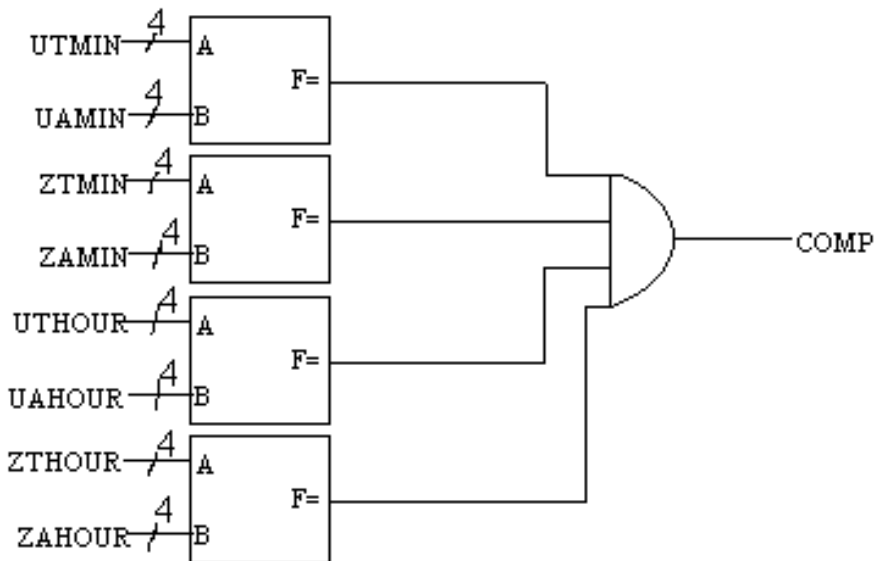


EN – SETALARM, CLK – CLK, M – M,

H – H, SET – VCC, RESET – RESET. Schema logică a modulului alarmei este:



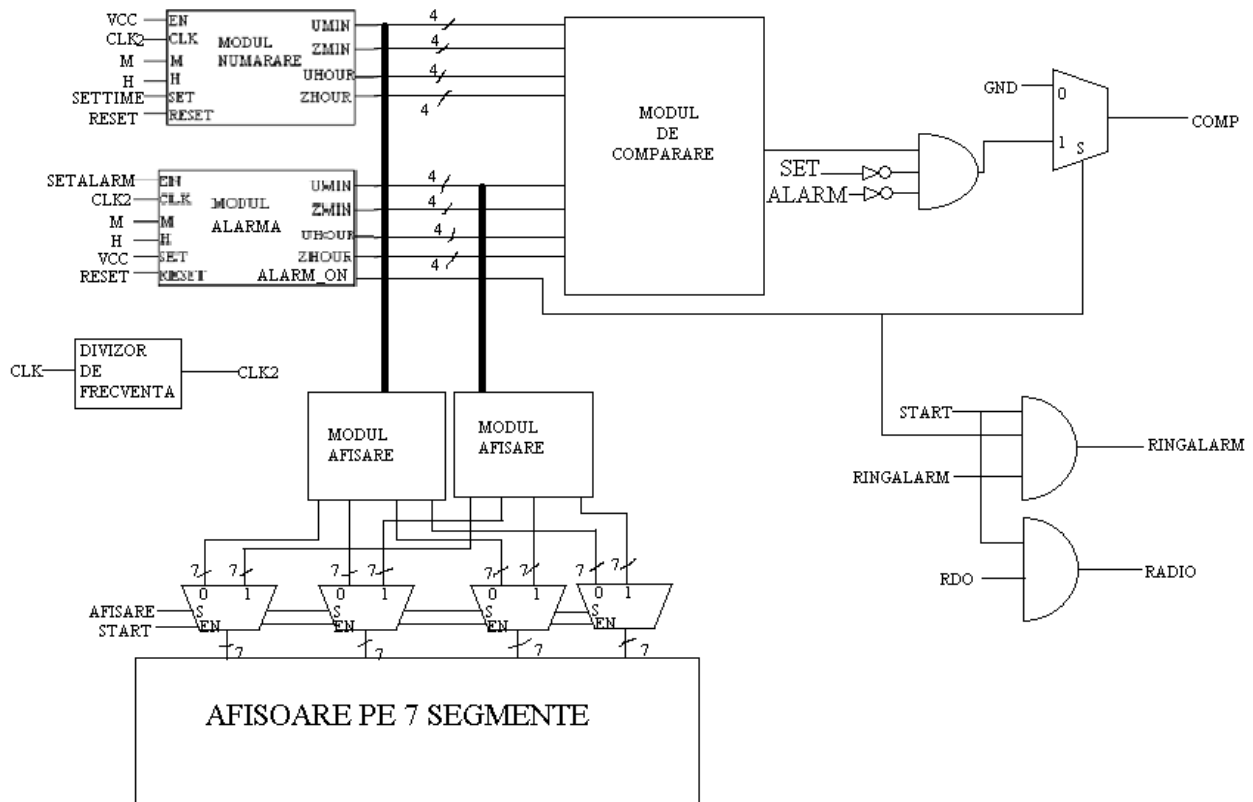
2. Modul de comparare: Va fi alcătuit din 4 comparatoare numerice pe 4 biți și va avea rolul de a compara datele curente cu datele setate pentru declanșarea alarmei.



3.Modul afișare: Va fi alcătuit din 4 decodificatoare 7 segmente.

4.Divizor de frecvență: Va fi alcătuit dintr-un numărător pe 2 biți și va avea rolul de a micșora frecvența semnalului de tact pentru a sincroniza afișarea pe afișoare și funcționalitatea componentelor aparatului.

Astfel, îmbinând toate modulele prezentate anterior, obținem schema finală a unității de execuție.



2.6 Sinteza Unității de comandă

Pentru început codificăm fiecare stare apoi realizăm diagrama Karnaugh pentru variabilele de stare viitoare în funcție de variabilele secundare curente. Registrul de stări interne se va realiza cu bistabile D flip-flop. Rezolvarea Diagramei Karnaugh duce la obținerea funcțiilor de excitație secundară pentru intrările D ale registrului de stări interne.

Funcțiile obținute în urma rezolvării diagramei Karnaugh sunt:

$$D_1 = ((\neg \text{STOPALARM} \bullet Q_1 \bullet Q_0 \bullet \text{ALARM}) + (\text{COMP} \bullet (\neg \text{ALARM}))) + \neg(\text{ALARM} \bullet \text{SET})$$

$$D_0 = ((\neg \text{STOPALARM} \bullet Q_1 \bullet Q_0 \bullet \text{SET}) + (\text{COMP} \bullet (\neg \text{SET}))) + \neg(\text{ALARM} \bullet \text{SET})$$

În continuare vom realiza diagrame Karnaugh pentru ieșiri pe baza organigramei.

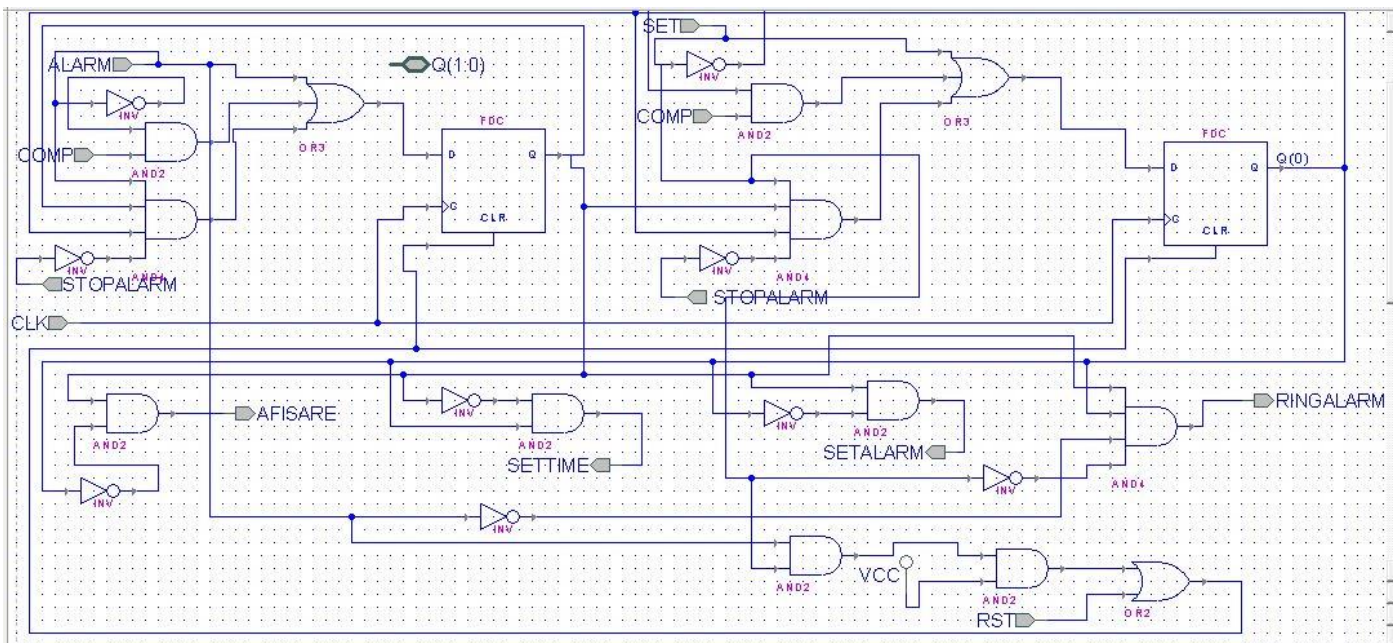
Funcțiile obținute în urma rezolvării diagramelor Karnaugh sunt:

$$\text{AFIȘARE} = \text{SETALARM} = Q_1 \bullet (\neg Q_0)$$

$$\text{SETTIME} = (\neg Q_1) \bullet Q_0$$

$$\text{RINGALARM} = (\neg \text{ALARM}) \bullet Q_1 \bullet Q_0 \bullet (\neg \text{SET})$$

Am ales implementarea funcțiilor de excitație/ieșire cu porți logice. Așadar schema organigramei devine:



3. Justificarea soluției alese

3.1 Alegerea resurselor

Numărătoarele au fost divizate în două părți (cifra unităților respectiv cifra zecilor) deoarece afișarea numerelor rezultate se va realiza pe afișoare pe 7 segmente. Această împărțire a numărătoarelor reprezintă o soluție rapidă și eficientă pentru acest tip de afișare.

Decodificatorul pentru afișajele pe 7 segmente va primi ca intrare o magistrală pe 4 biți. Deși rezultatul numărătoarelor diferă din punct de vedere al biților, am ales să alcătuiesc un singur decodificator cu intrarea pe 4 biți deoarece diferența de biți poate fi ușor completată prin adăugarea pe pozițiile cele mai semnificative a unor biți de '0' (*padding*).

Asemenea decodificatorului, comparatoarele sunt realizate și ele pe 4 biți. Deoarece în implementarea acestui aparat nu este relevantă decât funcția de egalitate, am ales să nu implementez funcția mai mare respectiv mai mic pentru a ușura și simplifica schema finală a aparatului.

În realizarea unității centrale am ales implementarea registrului de stări interne prin bistabile D, acestea fiind universal mai comune și simplu de implementat. Am ales implementarea ieșirilor prin porți logice fundamentale deoarece luând în considerare natura și complexitatea aparatului realizat, acestea reprezintă cea mai facilă și puțin costisitoare metodă de implementare.

3.2 Decizii de design

Deși esențiale în implementarea acestui aparat, am ales ca secunde să nu fie afișate pe afișoare pe 7 segmente deoarece am fost de părere că în citirea orei adesea secunde sunt prea puțin relevante iar majoritatea ceasurilor digitale le exclud. De asemenea, această decizie a dus la o schemă finală mai simplă și mai ușor de urmărit.

Am optat pentru implementarea a 2 butoane diferite pentru setarea alarmei respectiv oprirea alarmei pentru a oferi utilizatorului o imagine clară asupra funcționalității fiecărui buton.

Pentru a semnala faptul că alarma a fost setată, am ales ca în colțul drept-jos să apară un punct pentru a facilita interacțiunea utilizator-aparat.

4. Manual de utilizare și întreținere

4.1 Utilizarea aparatului

Pentru început vom conecta aparatul la o sursă de curent. Odată conectat, acesta va începe să funcționeze de la sine. Vom continua prin apăsarea butonului *START* care va activa toate funcțiile ceasului, inclusiv afișarea orei pe cele 4 afișoare. După pornirea efectivă a aparatului va fi necesar să setăm ora curentă. Pentru a face acest lucru, vom apăsa butonul *SET*, iar apoi prin apăsarea repetată a butoanelor *M* (minute) respectiv *H* (ore) vom incrementa ceasul până la ora dorită. După finalizarea incrementării, pentru a părăsi modul SETARE TIMP, vom apăsa din nou butonul *SET*. Ceasul va continua acum numărarea de la ora setată.

Pentru a seta o alarmă la o anumită oră, vom apăsa butonul *ALARM*, iar apoi echivalent cu metoda prezentată anterior pentru setarea ceasului, vom incrementa folosind aceleași butoane, ora și minutele. După finalizarea incrementării, pentru a părăsi modul SETARE ALARMĂ, vom apăsa din nou butonul *ALARM*. Alarma este acum setată, acest lucru fiind semnalat prin apariția unui punct în colțul drept-jos al afișajului. Acum ora curentă va fi iar afișată pe cele 4 afișaje. Când ceasul va ajunge la ora setată pentru alarmă, aceasta va fi declanșată timp de un minut, după care se va opri de la sine, însă va rămâne setată. **ATENȚIE!** Dacă alarma nu este oprită manual, aceasta va rămâne setată la aceeași oră pentru următoarea zi. Pentru a opri sau dezactiva alarma în mod manual, vom apăsa butonul *STOP ALARM*, care va opri și reseta alarma.

Pentru a porni radioul, trebuie să ne asigurăm că aparatul este pornit (butonul *START* a fost apăsat), iar apoi prin apăsarea butonului *RDO* îl vom activa. Pentru a opri radioul vom apăsa din nou același buton.

Butonul *RESET* va reseta atât ora curentă cât și alarma (în cazul în care a fost setată).

4.2 Semnificația butoanelor

Aparatul prezintă 8 butoane pentru manevrarea aparatului și 4 afișaje pe 7 segmente.

- Butonul *START* – pornește aparatul;

- Butonul *ALARM* – permite utilizatorului să seteze o alarmă la o oră dorită;
- Butonul *SET* – permite utilizatorului să seteze ceasul la o oră dorită;
- Butonul *STOP ALARM* – permite utilizatorului să oprească/dezactiveze alarma;
- Butonul *RESET* – reinițializează aparatul;
- Butonul *RDO* – permite utilizatorului să pornească/oprească radioul;
- Butonul *M* – utilizat în incrementarea minutelor la setarea ceasului/alarmei;
- Butonul *H* – utilizat în incrementarea orelor la setarea ceasului/alarmei.

4.3 Administrarea aparatului și alte atenționări

În cazul unei funcționări neadecvate a aparatului se recomandă apăsarea butonului *RESET* pentru a reinițializa echipamentul la starea originală.

ATENȚIE! Aparatul funcționează atât timp cât este conectat la o sursă de curent, însă dacă butonul *START* nu este apăsat acesta nu va afișa nimic pe cele 4 afișaje iar alarma nu se va declanșa. De asemenea, radioul nu va porni decât dacă butonul *START* este apăsat.

5. Posibilități de dezvoltare

Având în vedere că aparatul poate fi setat la o anumită oră prin incrementare manuală, prin înlocuirea numărătoarelor cu numărătoare reversibile și prin adăugarea a încă 2 butoane (decrementarea minutelor respectiv decrementarea orelor) setarea orei ar putea fi mai facilă oferind utilizatorului posibilitatea de a decrementa orele/minutele și nu doar incrementa.

Bibliografie

https://en.wikipedia.org/wiki/Seven-segment_display#Implementations

<https://www.electronics-tutorials.ws/blog/7-segment-display-tutorial.html>

https://www.tutorialspoint.com/computer_logical_organization/digital_counters.html

Proiectare Logică, Îndrumător de laborator

C. Vancea, **Cursuri Proiectare Logică**, 2020-2021

O. Creț, L. Văcariu, **Probleme de Proiectare logică**, U.T. PRESS 2008 Cluj-Napoca