

5.6. Generator de sunete

Acest modul echipează MC pe 16 biți (MB90390, 90420, 90590) și MC pe 32 de biți (MB91360), structura și funcționarea fiind identice.

Principiul generării sunetelor

Printr-o primă metodă sunetele se obțin dintr-un semnal dreptunghiular cu frecvența dorită a sunetului generat, și un generator PWM care, prin modificarea factorului de umplere asigură un volum variabil, schema și formele de undă fiind date în figura 5.49:

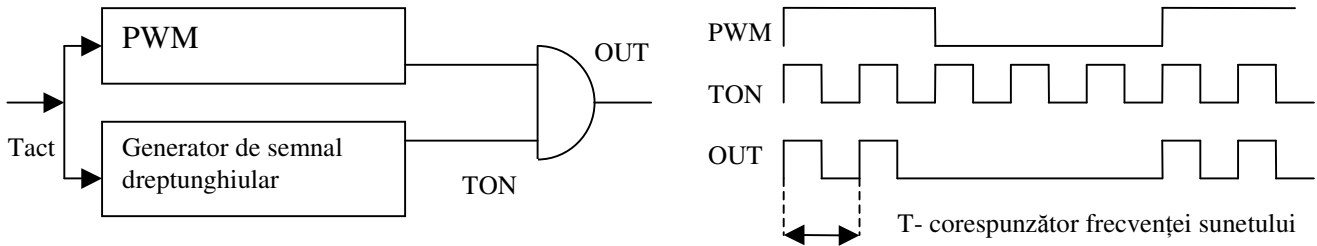


Figura 5.49 : metodă simplă de a genera sunete

Metoda folosită în modulul de generare sunete în MC este ceva mai complexă, asigură mai multe facilități iar sunetul generat este mai bun. Modificarea față de metoda precedentă este că generatorul PWM constituie tact pentru generatorul de frecvență a sunetului, figura 5.50.

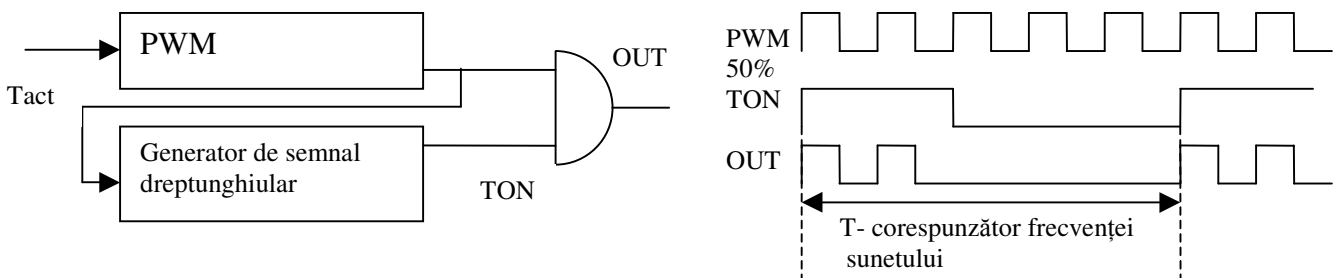


Figura 5.50: metoda folosită de generatorul de sunete integrat

Modificarea volumului se obține tot prin modificarea factorului de umplere, ca în diagramele din figura 5.51:

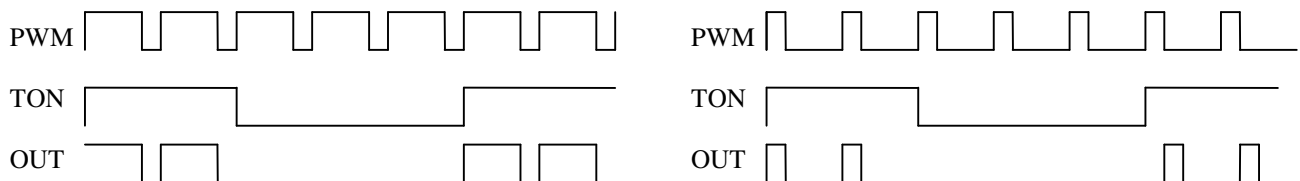


Figura 5.51: modificarea volumului

În diagrama din stânga s-a comandat un volum mai mare decât în diagrama din dreapta. Semnalul de ton este semnalul care stabilește frecvența sunetului generat. Acest semnal trebuie să aibă un factor de umplere de 50%.

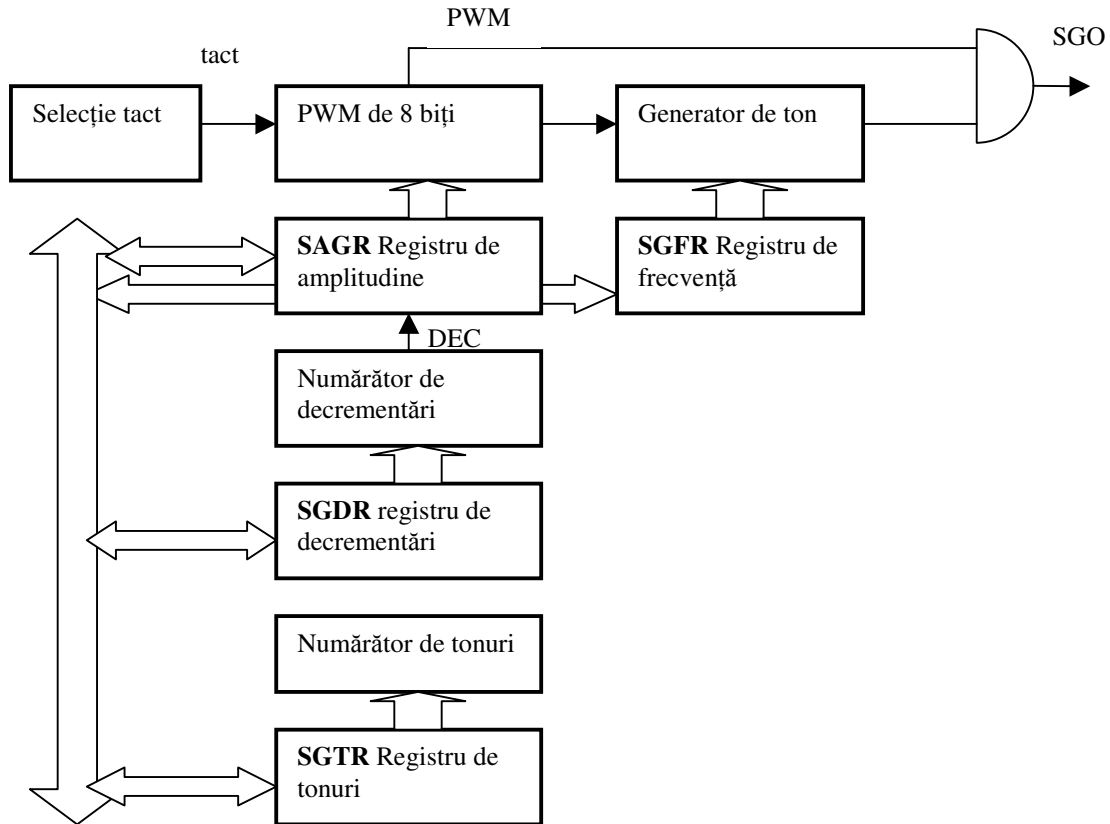
Schema bloc a generatorului de sunet (figura 5.52)


Figura 5.52: schema bloc a generatorului de sunet

Regiștrii și funcționarea generatorului de sunet

SGCR (Sound Control Register), un bit arată că generatorul operează, un bit validează micșorarea volumului, 2 biți stabilesc tactul (tact pentru periferice la MB91360 să tact mașină pentru MB90xxxx, divizat cu 1, 2, 4 sau 8), un bit stabilește dacă semnalul de ieșire este o formă simplă dreptunghiulară sau un ȘI cu semnalul PWM, un bit validează întreruperile, un bit cere întrerupere după generarea numărului de tonuri programate, un bit pornește generatorul de sunete.

SGFR (Frequency Data Register) stochează valoarea de reîncărcare pentru numărătorul care stabilește frecvența semnalului de ton, figura 5.53:

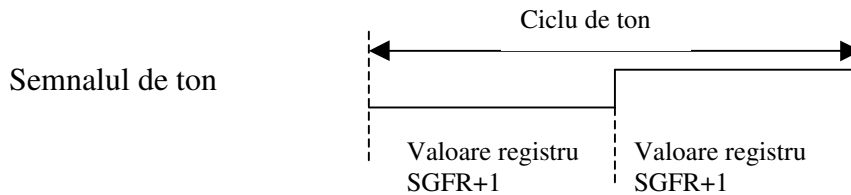


Figura 5.53: semnalul de ton

SAGR (Amplitude Data Register) stochează valoarea de reîncărcare pentru generatorul PWM și reprezintă amplitudinea sunetului. Valoarea se reîncarcă în generatorul PWM la sfârșitul fiecărui ciclu de ton. În figura 5.54 sunt reprezentate 3 exemple de diagrame pentru 3 valori diferite de reîncărcare:

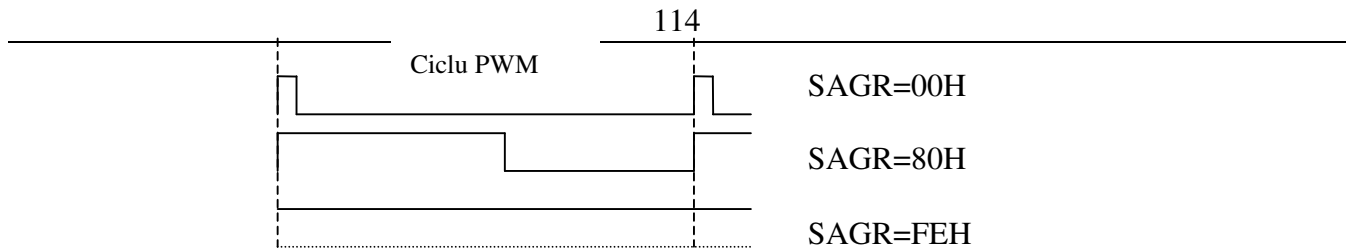


Figura 5.54: diagrame de timp pentru 3 factori de umplere diferiți

SGDR (Decrement Grade Register) valoarea din SAGR este decrementată după fiecare reîncărcare (dacă micșorarea volumului este validată). Micșorarea automată a volumului are avantajul că UC intervine puțin în cadrul acestei operații. Decrementarea lui SAGR se face cu numărul specificat în SGDR+1.

SGTR (Tone Count Register) stochează valoarea de reîncărcare pentru numărătorul de impulsuri de ton.

SGBDL (Sound Disable Register) cu un bit validează/ invalidează generatorul.

În cazul selectării ieșirii de ton ca ieșire SGO, forma generată va fi o formă simplă dreptunghiulară, figura 5.55:

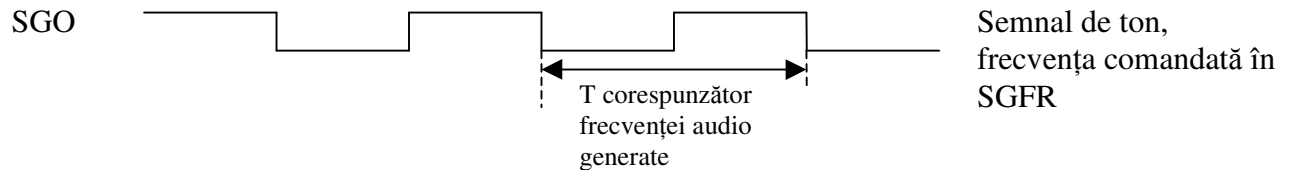


Figura 5.55: ieșire simplă dreptunghiulară (ton continuu)

Dacă se selectează ieșirea SGO ca ieșire combinată între PWM și semnalul de ton (ȘI logic) se obține, figura 5.56:

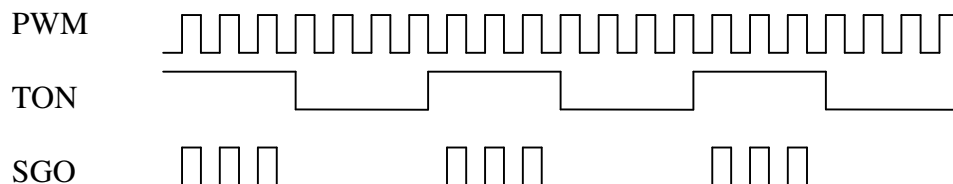


Figura 5.56: obținerea variației volumului

Dacă se modifică factorul de umplere al PWM în SAGR se obține un volum variabil.

Micșorarea automată a volumului se realizează prin micșorarea factorului de umplere, conform cu diagrama din figura 5.57.

În această diagramă se vede micșorarea volumului în 3 trepte. La terminarea decrementării SAGR se generează o întrerupere.

Frecvența minimă audio care se poate genera se apreciază dacă considerăm factorul maxim de divizare ($24\text{MHz}/8 = 3\text{MHz}$) și divizarea prin 2 regiștrii de 8 biți (PWM și generatorul de frecvență) ceea ce înseamnă circa 45Hz.

Dacă este validată micșorarea automată a volumului, în SGTR se stochează valoarea de reîncărcare pentru numărătorul de tonuri sau numărul de decrementări. Dacă în diagrama anterioară s-a definit SGTR=3, după cele 3 decrementări (valoarea cu care se decrementează este stocată în SGDR), se activează o cerere de întrerupere. În această întrerupere se poate opri sunetul sau se poate genera un alt sunet.

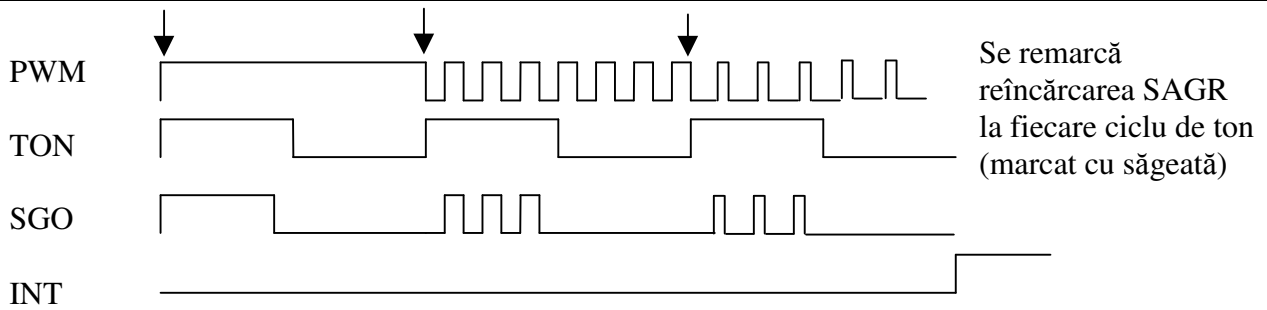


Figura 5.57: micșorarea automată a volumului

5.7.Modul pentru comanda motoarelor

MC Fujitsu sunt folosite mult în aplicații auto, de aceea se integrează un modul dedicat comenzii motoarelor pas cu pas, dar care poate comanda și motoare de curent continuu. Acest modul apare deja la MC pe 8 biți de tip vechi (MB89940), echipează multe MC pe 16 biți (MB90390, 90420, 90590, 90595) și MC pe 32 de biți (MB91360).

Modulul de comandă pentru comanda motoarelor conține 2 generatoare PWM, 4 etaje de ieșire de putere (drive de motor) și logica de selecție. Driverul poate asigura un curent mare care permite conectarea directă a motoarelor de mică putere. Un mecanism de sincronizare asigură funcționarea sincronizată a celor 2 generatoare PWM. În general MC conțin mai multe asemenea module.

Schema bloc a unui modul de control motoare care conține 2 canale PWM și 4 drive de ieșire este dată în figura 5.58:

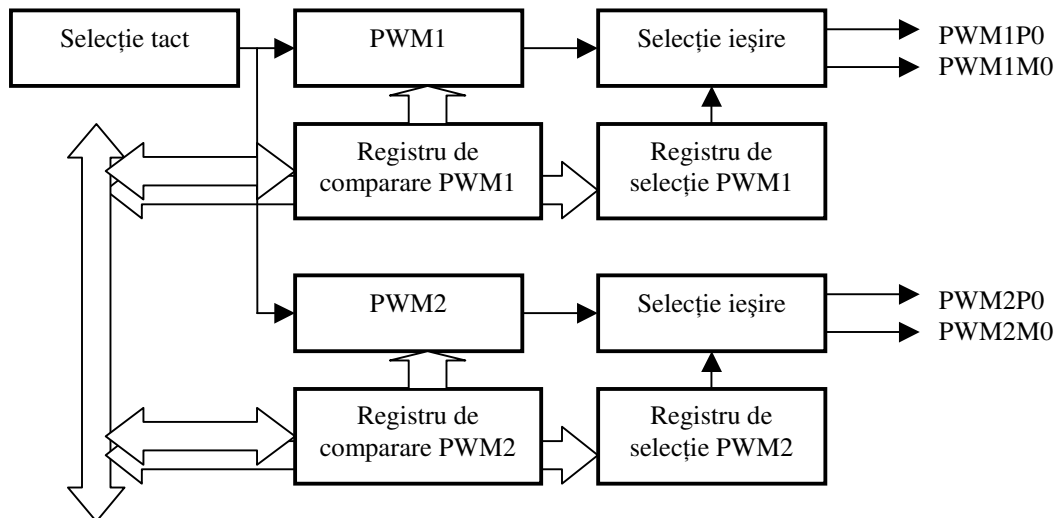


Figura 5.58: schema bloc a modului de comandă a motoarelor

Numărătorul PWM pentru fiecare canal numără până la capăt, apoi se reîncarcă și numărarea reîncepe. La coincidența valorii din numărător cu cea stocată în registrul de comparare semnalul de ieșire își schimbă starea logică. La sfârșitul numărării semnalul de ieșire își schimbă din nou starea logică.

Registrii

PWM Control Register (8 biți) controlează pornirea/ oprirea operării, pinii externi și întreruperile. 2 biți selectează funcționarea pinilor externi ca pini de control motor sau pini de I/O de uz general, 2 biți selectează tactul (tact divizat cu 1, 2, 4 sau 8), un bit pornește operarea (PWM2 pornește

un tact după PWM1 pentru a micșora zgomotul produs în driver la o comutare simultană de curenți mari), un bit selectează operarea PWM pe 8 sau 10 biți.

PWM1/2 Compare Register (16 biți) specifică lățimea impulsului PWM (formatul de scriere este pe 8 sau 10 biți). O valoare de 000H indică un factor de umplere de 0% iar o valoare de 3FFH indică un factor de umplere de 99,9% (99,6% pentru formatul pe 8 biți). Pentru a putea stoca valori pe 10 biți regiștrii de comparare sunt pe 16 biți. 3 variante de semnale cu factor de umplere diferit sunt arătate în figura 5.59:

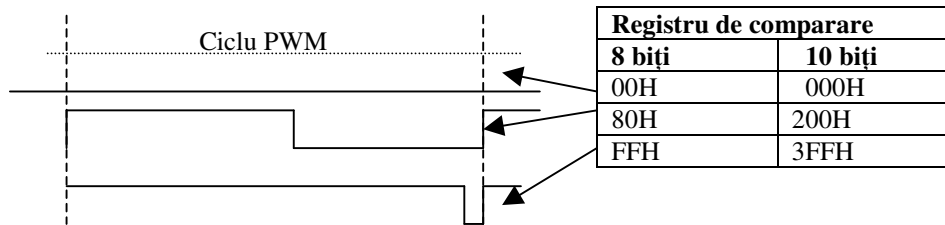


Figura 5.59: 3 diagrame de timp pentru 3 factori de umplere diferiți

PWM1/2 Selection Register controlează pinii de ieșire. Un bit asigură sincronizarea ieșirilor în sensul că orice modificare în configurarea canalelor PWM nu se reflectă la ieșire decât după ce acest bit este setat. Câte 3 biți selectează tipul de semnal de ieșire la ieșirile PWM1P0, PWM1M0, PWM2P0, PWM2M0- semnal în stare L, H, impedanță ridicată sau PWM.

Comanda motoarelor de curent continuu

Cuplând un motor de curent continuu între PWM1P0 și PWM2P0 se poate obține o rotație cu viteză variabilă în ambele sensuri ale motorului. Motorul de curent continuu se rotește cu o viteză proporțională cu tensiunea aplicată. Dacă tensiunea aplicată este sub formă de impulsuri motorul se rotește proporțional cu valoarea medie a tensiunii. În diagrama următoare (figura 5.60) se exemplifică această comportare în cazul rotirii cu viteză maximă într-un sens sau în altul, apoi cu se arată o diagramă de semnal pentru comanda în caz general:

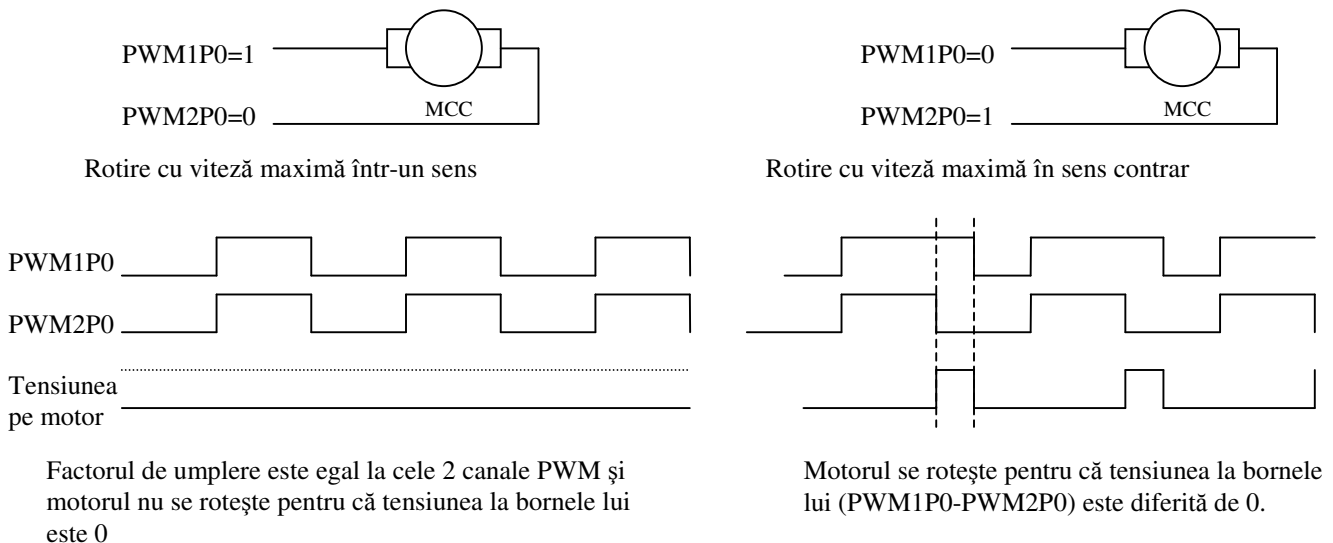


Figura 5.60: comanda unui motor de curent continuu

O simulare în SIMULINK dovedește valabilitatea acestei metode de control. S-a folosit un model din SIMULINK pentru motorul de curent continuu și cele 2 canale PWM au fost simulate cu generatoare de impulsuri cu lățime variabilă, figura 5.61:

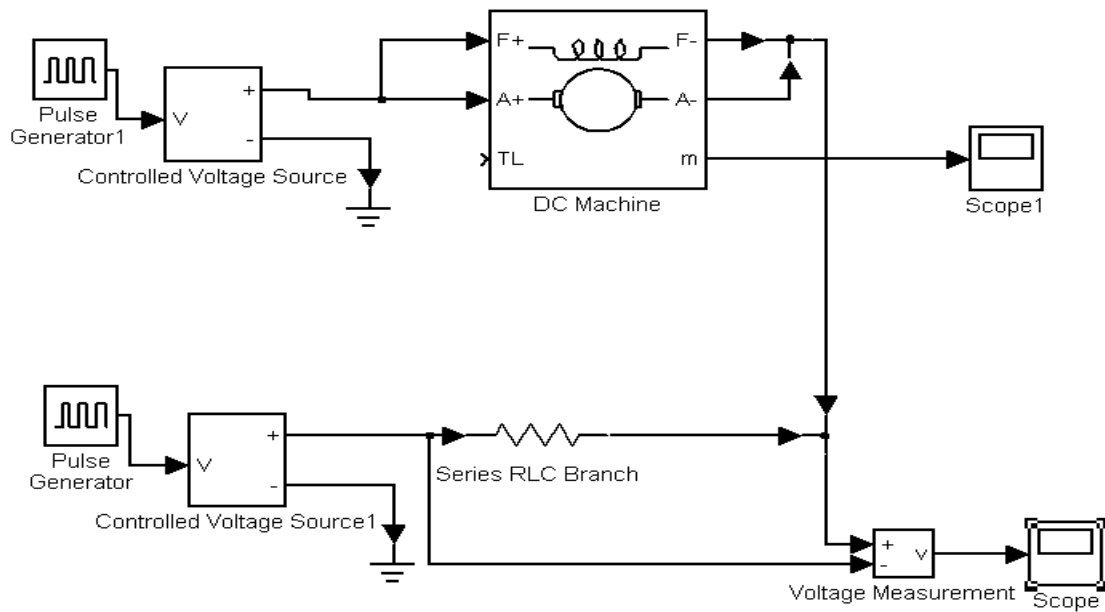


Figura 5.61: model SIMULINK pentru simularea funcționării motorului de curent continuu

Rezultatele simulării sunt reprezentate în figura 5.62. În stânga sus este reprezentată forma curentului prin motor și jos turația obținute pentru factor de umplere de 90% respectiv 10%. Se vede că acestea au aceeași formă de variație. În dreapta sunt reprezentate aceleași curbe, dar cu factor de umplere 10% respectiv 90%. Se vede că motorul se rotește în sens contrar.

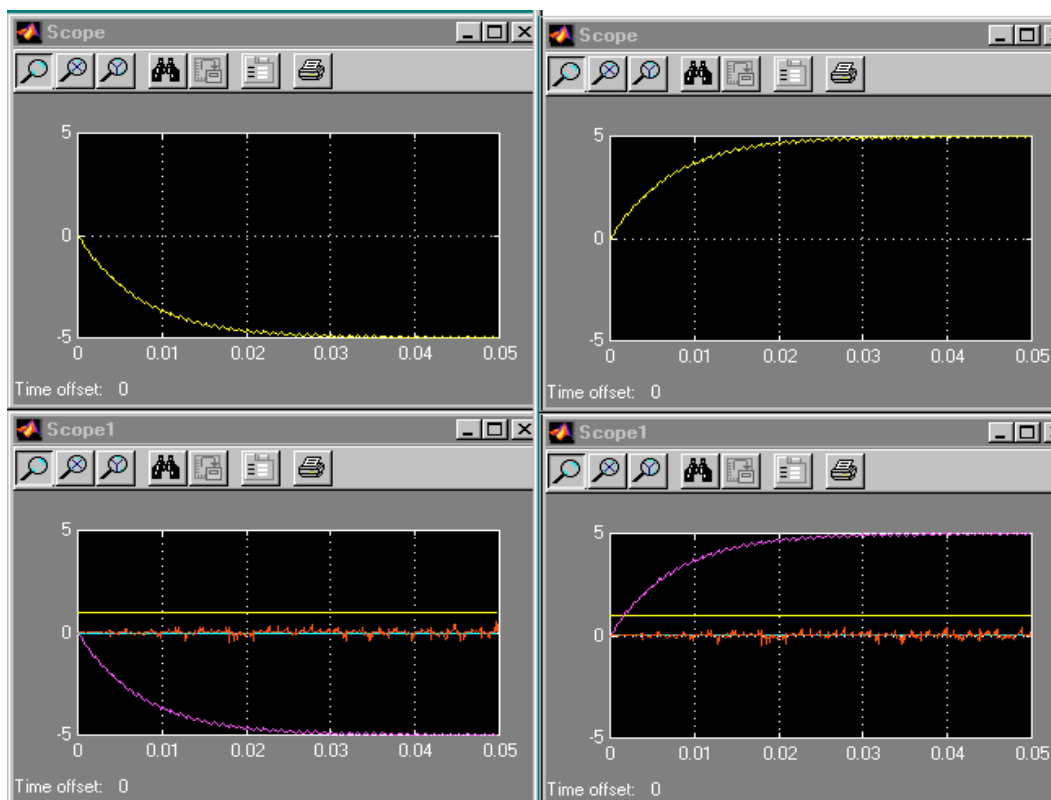
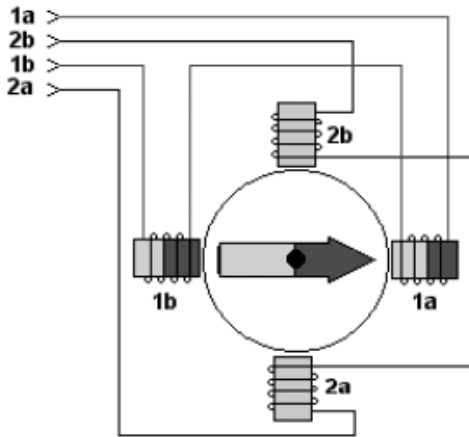


Figura 5.62: rezultatele simulării SIMULINK

Comanda motoarelor pas cu pas

Motoarele pas cu pas pot fi motoare unipolare sau bipolare. La motoarele bipolare comanda pașilor se face prin inversarea curentului prin înfășurări. Principiul comenzii seamănă cu cel de la comanda motorului de curent continuu, cu diferența că în acest caz de regulă sunt 2 înfășurări, figura 5.63:



În acest caz controllerul trebuie să poată inversa polaritatea pentru o deplasare a curentului în ambele sensuri. Controllerul trebuie să alimenteze înfășurările succesiv cu o anumită secvență pentru un sens și secvența inversă pentru celălalt sens. În desenul alăturat o parcurgere a 4 faze înseamnă o rotire de 360° .

Figura 5.63: structura motorului pas cu pas bipolar

Conectarea unei înfășurări (de exemplu 1a-1b) la cele 2 canale PWM permite inversarea curentului ca în diagramele din figura 5.64:

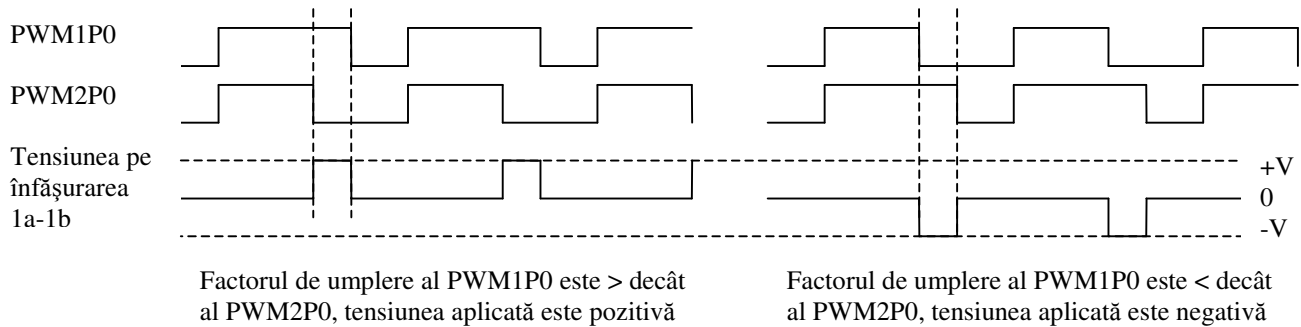
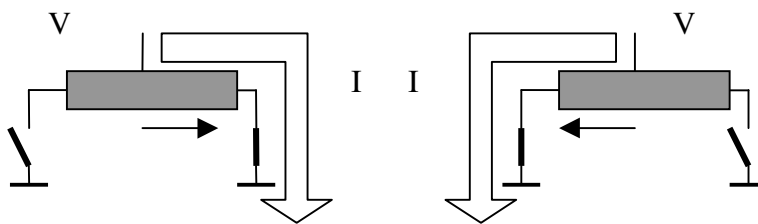


Figura 5.64: inversarea curentului în înfășurări la motoarele bipolare

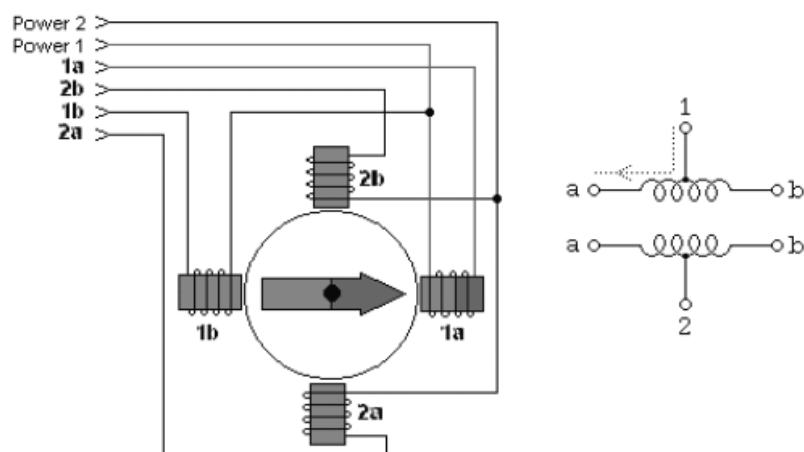
MPP unipolare folosesc o priză mediană legată la alimentare, inversarea curentului obținându-se prin legarea la masă succesivă a terminalelor extreme ale înfășurării, ca în figura 5.65:



Se obțin astfel câmpuri magnetice de sens contrar fără inversarea polarității

Figura 5.65: inversarea curentului în motoarele unipolare

Înfășurările motorului sunt conectate ca în figura 5.66:



Prizele mediane sunt legate la alimentare și secvența de impulsuri se aplică terminalelor 1a, 1b, 2a, 2b.

Figura 5.66: structura și înfășurările motorului unipolar

Observație

O perioadă a semnalului PWM nu este semnalul pentru efectuarea unui pas. Semnalul PWM nu asigură succesiunea fazelor ci valoarea medie a tensiunii de alimentare a unei faze. Conectarea ieșirilor PWM la fazele unui motor unipolar este dată în figura 5.67:

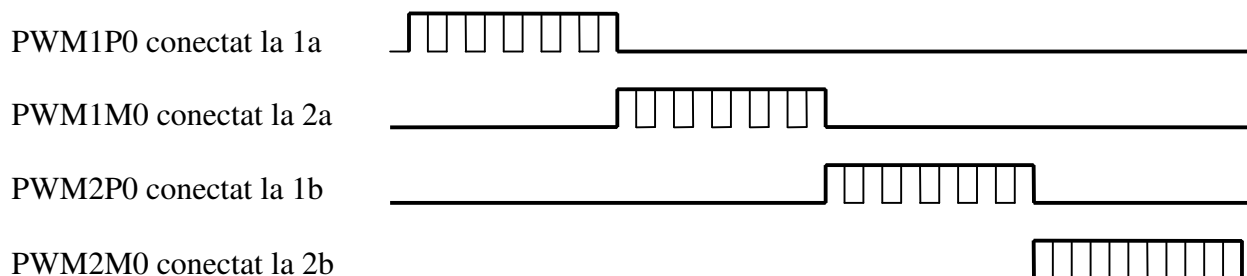


Figura 5.67: comanda fazelor unui motor unipolar

Un modul de comandă poate controla un motor pas cu pas asigurând un curent bine definit prin înfășurări. Acest tip de comandă crează de exemplu posibilitatea de a comanda motorul cu un curent mai mare la pornire sau permite realizarea unor traiectorii optime de viteză (regim accelerat- frânat).

Detectorul de poziție zero

Modulul de comandă al motoarelor din componența MC pe 32 de biți (MB91360) are în plus un bloc de detectare a poziției de zero a rotorului¹ care pune la dispoziție informații despre poziția rotorului.

Schema bloc a blocului este dată în figura 5.68.

Logica de analiză eșantionează ieșirea comparatorului cu tactul selectat în ZPD0. Comparatorul compară intrarea de la PWM2M0 cu tensiunea de referință și setează bitul de rezultat al comparației dacă tensiunea de intrare este mai mare decât tensiunea de referință. Rezultatul comparației devine 1

¹ Această funcție a controllerului este patent al Siemens VDO Automotive AG și poate fi folosită doar cu acordul proprietarului.

dacă toate eșantioanele (numărul lor este specificat în ZPD0) sunt 1. Ca urmare pinul PWM2M0 poate fi folosit pentru închiderea unei bucle de reacție de control a motorului.

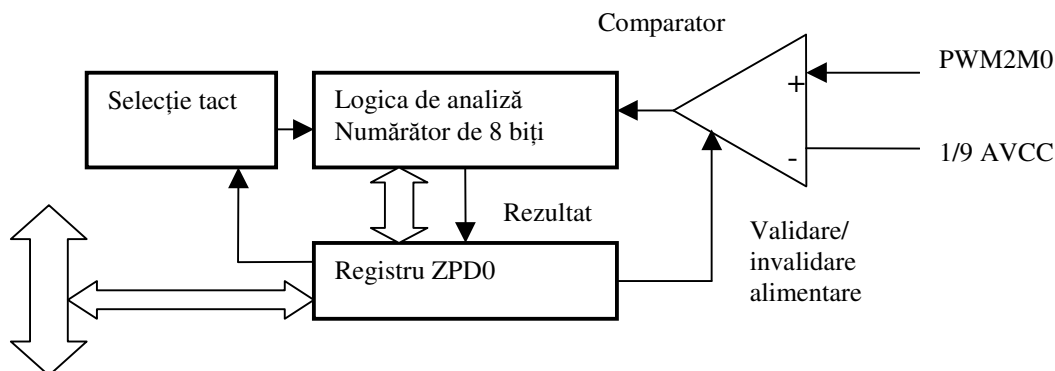


Figura 5.68: schema bloc a detectorului de poziție zero

Registru ZPD0 (Zero Detect Register) conține: 3 biți selectează tactul (tact divizat cu 1,2,4,5,6 sau 8), un bit validează operarea detectorului de zero, 3 biți stabilesc numărul de eșantioane (1,2,3,4 sau 5), un bit validează/ invalidează alimentarea blocului analogic, un bit indică rezultatul comparării.

5.8. Controller LCD

Controllerul LCD (Liquid Crystal Display) dispune de o memorie internă video de 20 de octeți și comandă un display LCD prin intermediul a 4 ieșiri comune și 40 de ieșiri de segment (modelul descris este cel din familia MB95xxxx). Pot fi selectate 3 variante de multiplexare (duty). Controllerul afișează conținutul memoriei direct pe LCD. Unele controllere dispun de funcția hardware de clipire. Pini utilizați de controllerul LCD sunt:

- COM0/3 4 pini pentru ieșirile comune
- SEG0/39 40 de pini pentru comanda segmentelor LCD
- V0/3 pini de ieșire pentru tensiunile de comandă a LCD sau pini de conectare a capacitivelor, funcție de sistemul de alimentare
- C0, C1 pini pentru conectarea capacitivelor la modelele cu sursă internă.

Principiul de comandă al LCD

Un pixel (sau o zonă a display-ului, segment la o cifră reprezentată cu 7 segmente sau o formă specială) este aprins dacă diferența de potențial între borna comună și cea de segment este maximă (starea ON). Aprinderea prin aplicarea de impulsuri este necesară ținând cont de structura chimică a LCD care se deteriorează dacă este comandat în curent continuu.

Pentru o comandă cu 2 intervale de tensiune (bias=1/2) nemultiplexat (afișare statică) (duty=1) conexiunile la o cifră reprezentată cu 7 segmente sunt arătate în figura 5.69:

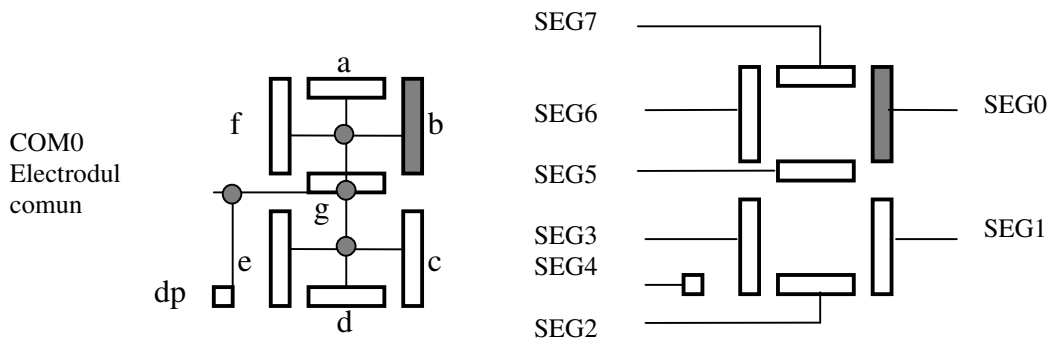


Figura 5.69: conexiuni pentru comanda nemultiplexată

Diferențele de potențial între electrozii comuni și cei de segment determină aprinderea segmentelor astfel:

- COM0-SEG0 comandă segmentul b
- COM0-SEG1 comandă segmentul c
- COM0-SEG2 comandă segmentul d, etc.

Diagramele de timp din figura 5.70 arată formele de undă aplicate la 2 segmente:

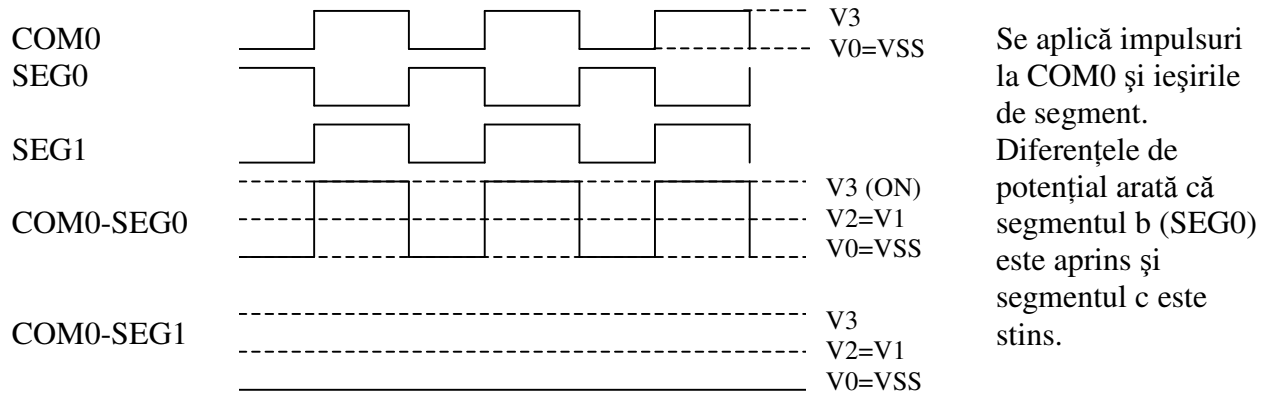


Figura 5.70: diagrame de timp pentru o comandă nemultiplexată cu 2 intervale de tensiune

În realitate, pentru a evita comanda segmentului c cu nivel continuu, nivelele de tensiune ale COM0 și SEG0/1 se stabilesc și la valorile intermediare V2 sau V1, lucru care se va clarifica la comanda cu 3 nivele de tensiune.

Pentru comanda multiplexată pe 2 căi (duty=1/2) conexiunile sunt arătate în figura 5.71:

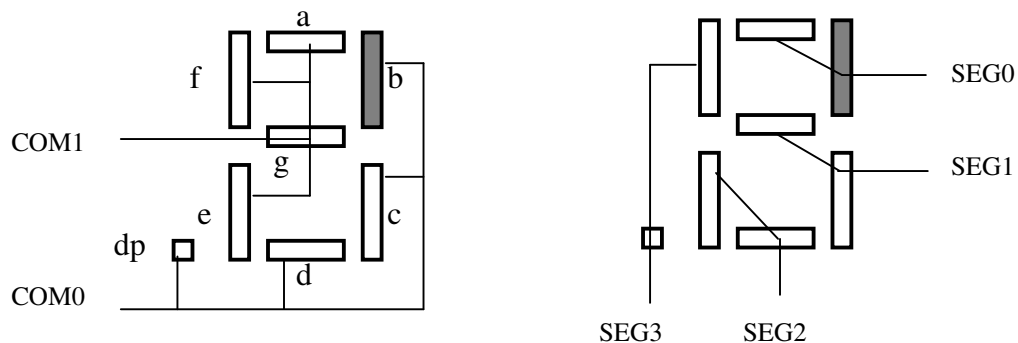


Figura 5.71: conexiuni pentru comanda multiplexată pe 2 căi

Comanda multiplexată pe 2 căi înseamnă folosirea a 2 terminale comune iar pe 4 căi înseamnă folosirea a 4 terminale comune ($\text{duty}=1/4$). Nu sunt posibile toate combinațiile între numărul de intervale de tensiune și multiplexări. Variantele posibile în funcție de modul de alimentare (cu sursă internă sau fără sursă internă) sunt date în foile de catalog.

Diferențele de potențial între electrozii comuni și cei de segment determină aprinderea segmentelor astfel:

- COM0-SEG0 comandă segmentul b
- COM0-SEG1 comandă segmentul c
- COM1-SEG0 comandă segmentul a
- COM1-SEG1 comandă segmentul g, etc.

Diagramele de timp pentru comanda multiplexată pe 2 căi și 3 intervale de tensiune ($\text{bias}=1/3$) sunt date în figura 5.72:

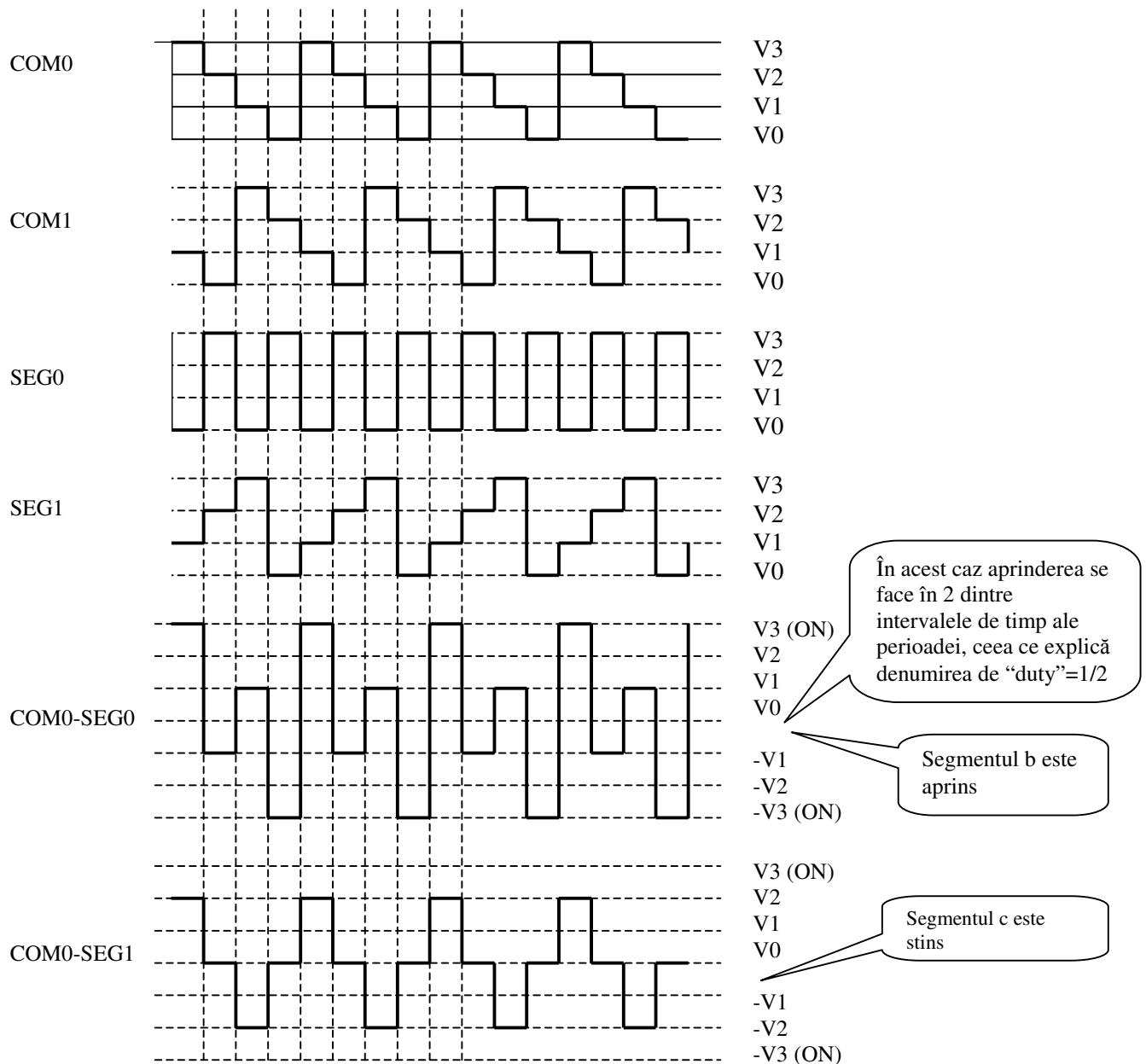


Figura 5.72: diagrame de timp pentru comanda multiplexată pe 2 căi și 3 intervale de tensiune

Schema bloc a controllerului LCD este dată în figura 5.73:

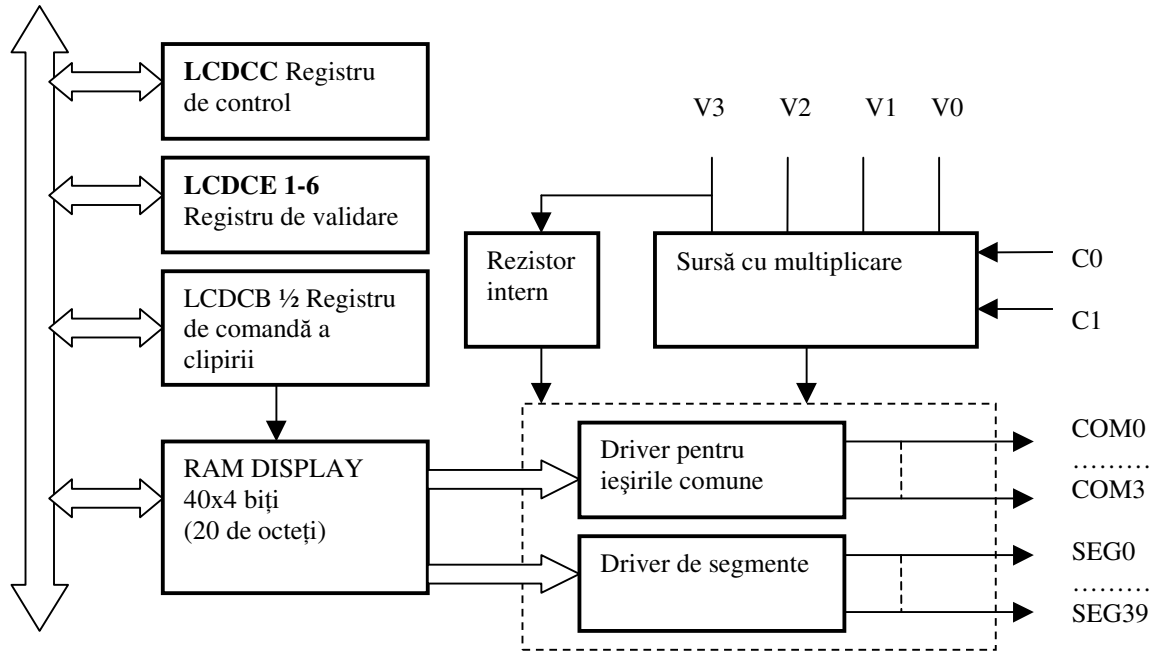
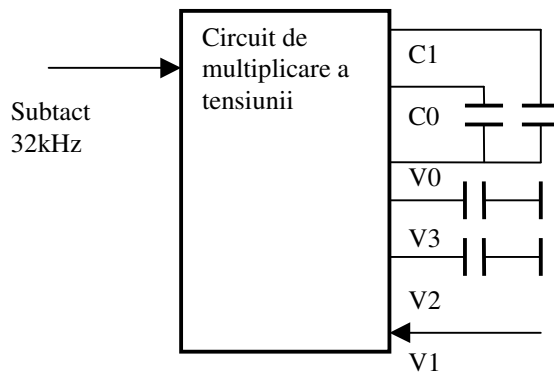


Figura 5.73: schema bloc a controllerului LCD

Moduri de alimentare

1. Există modele de MC cu sursă internă de multiplicare a tensiunii (step-up), doar la modelele cu sistem dual de tact, figura 5.74:



La pinul V1 se aplică tensiunea de referință și la pinii C1, C0, V0, V3, V2 se conectează condensatori.

Figura 5.74: alimentare cu sursă internă cu multiplicare

2. Varianta cu divizor poate fi cu divizor intern și atunci conține o rezistență ca divizor sau cu divizor extern și atunci se poate conecta o rezistență externă de la V0 la V3. Aceste rezistențe sunt comandate de bitul VSEL din registrul LCDCC, astfel dacă VSEL=1 se folosește divizorul intern. La modelele care nu au divizor intern se conectează rezistențe în exterior. Schemele bloc pentru variantele cu divizor sunt date în figura 5.75.

La alimentarea cu divizor extern dacă rezistența între V2 și V1 există se obțin 3 intervale de tensiune (bias=1/3). Dacă rezistența nu există se obțin 2 intervale de tensiune (bias=1/2). V_{LCD} este tensiunea de operare pentru LCD. Rezistențele externe depind de tipul de LCD folosit.

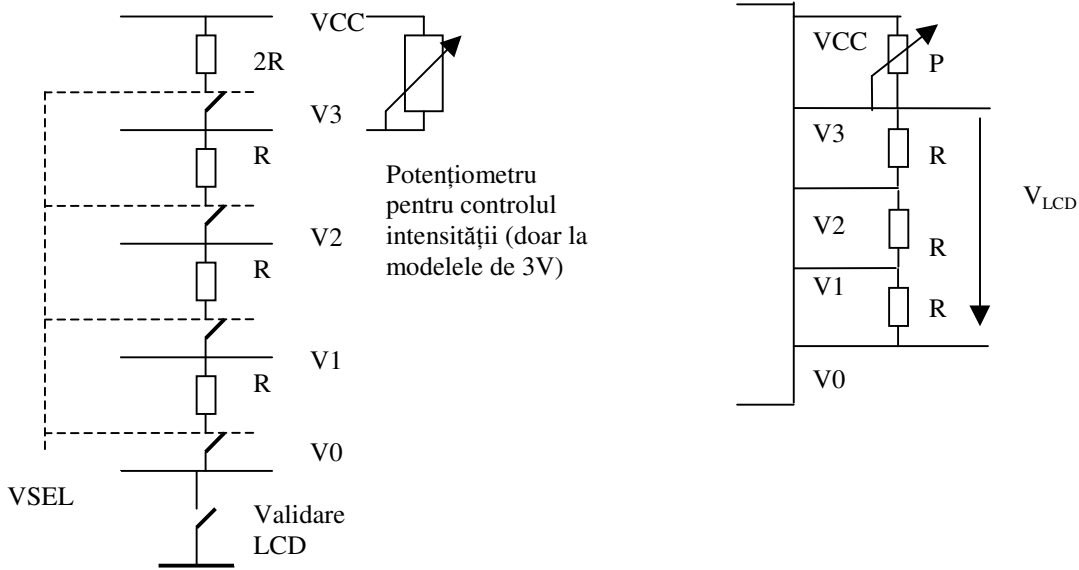


Figura 5.75: alimentarea cu divizor intern (stânga) și extern (dreapta)

Regiștrii

LCDDC (LCD Control Register) este folosit pentru a selecta tactul, modul de afișare și modul de alimentare. Un bit selectează sursa tactului (timerul de bază, deci oscilatorul principal sau timerul de supraveghere, deci subtactul), un bit stabilește dacă afișajul continuă să opereze la intrarea în mod stop sau la schimbarea tactului de la tact principal la subtact, un bit validează funcționarea rezistorului intern de divizare, 2 biți selectează modul de multiplexare și 2 biți stabilesc factorul de divizare al tactului.

LCDE1 (LCD Enable Register 1) controlează pini externi și perioada de clipire. Un bit selectează funcția pinilor ca pini de comandă LCD sau pini de intrare (ca pini de intrare se micșorează curentul consumat), un bit selectează perioada de clipire, câte un bit selectează funcțiile pentru V1, V2, V3, COM1, COM2, COM3, ca pini de comandă LCD sau pini de uz general.

LCDE2-6 (LCD Enable Register 2-6) stabilesc cu câte un bit dacă ieșirea SEG este de comandă a LCD sau pin de I/O de uz general.

LCDCB1/2 (LCD Blinking Setting Register). Clipirea se aplică segmentelor specificate de combinația între SEG0-SEG3 până la COM0-COM3. Fiecare bit stabilește dacă combinația SiCj (i-numărul segmentului între 0 și 3 și j numărul terminalului comun între 0 și 3) determină un punct clipitor (dacă bitul corespunzător este 1) sau un punct care nu clipește.

Memoria RAM de afișare este citită sincronizat cu activarea segmentelor și a terminalelor comune. Memoria poate fi scrisă oricând. Corespondența între structura RAM și terminale este, figura 5.76:

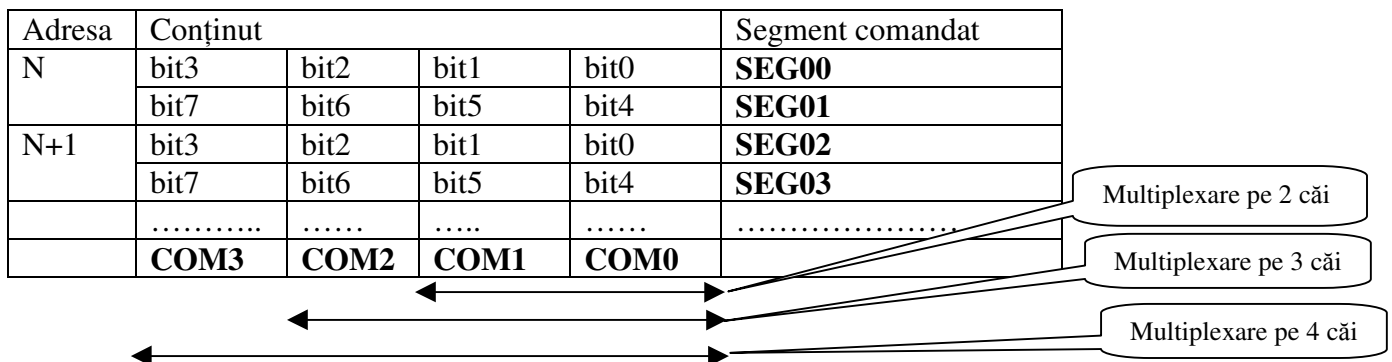


Figura 5.76: maparea memoriei video LCD

5.9. Magistrala CAN

5.9.1. Aspecte teoretice

CAN (Controller Area Network) este o magistrală serială introdusă de Bosch. Transmisia se face pe 2 fire, half duplex, cu mare viteză, în condiții de siguranță mare de funcționare. CAN poate lega teoretic 2032 dispozitive dar practic se pot conecta maximum 110 noduri. Modurile de detectare și tratare a erorilor fac ca această magistrală să fie preferată pentru medii perturbate sau pentru aplicații auto, unde funcționarea sigură este importantă.

CAN folosește protocolul CSMA/CD+AMP (Carrier Sense Multiple Access/ Collision Detection + Arbitration on Message Priority). Înainte de a trimite un mesaj transmițătorul CAN verifică dacă magistrala este liberă, începe să transmită, apoi verifică existența unei coliziuni. Același principiu este aplicat și la Ethernet, unde ca reacție la apariția unei coliziuni transmițătoarele abandonează transmisia și reîncearcă după perioade aleatoare de timp, ceea ce face ca transmisia să aibă probleme la un număr mare de dispozitive cuplate pe linie. La CAN este aplicat un alt principiu de reacție după detectarea unei coliziuni. Fiecare mesaj CAN conține un identificator (11 sau 29 biți) și toate nodurile receptoare recepționează mesajul și îi aplică un test de acceptanță. Dacă identificatorul este recunoscut ca relevant mesajul este procesat, dacă nu este ignorat. Prioritatea unui mesaj este dată de identificator, cu cât identificatorul reprezintă un număr mai mic, cu atât prioritatea este mai mare. Mesajele cu prioritate mai mică așteaptă eliberarea magistralei pentru a fi trimise.

Linia de transmisie fiind unică pentru toate nodurile (dispozitivele cuplate), nivelul de 0 se numește dominant și nivelul 1 recesiv. Dacă 2 noduri transmit simultan unul valoarea 0 și unul valoarea 1, linia trece în 0 (de aceea nivelul 0 se numește dominant). În acest mod se poate detecta o coliziune, când un transmițător transmite un nivel de 1 și citește pe linie un nivel de 0 înseamnă că mai multe transmițătoare transmit simultan. În acest caz transmițătorul cu prioritate mai mică abandonează transferul.

Două noduri nu pot transmite mesaje cu același identificator. Dacă acest lucru se întâmplă totuși și transmiterea este simultană nu se poate detecta coliziunea. Coliziunea este detectată mai târziu, în câmpul de date și se utilizează mecanismul de gestionare a erorilor CAN și nu mecanismul de arbitrare.

Cadrele CAN pot fi de 2 feluri:

- Cadre de date
- Cadre de cerere de informație (Remote Frame). Un nod poate cere informații transmițând un astfel de cadru. Nodul care dispune de această informație trebuie să răspundă.

Magistrala CAN poate fi:

- CAN standard (Versiunea 2.0 A) cu identificator pe 11 biți
- CAN extins (Versiunea 2.0 B) cu identificator pe 29 de biți.

Cadrul de date CAN 2.0A

Structura unui cadru CAN 2.0A este dată în figura 5.77:

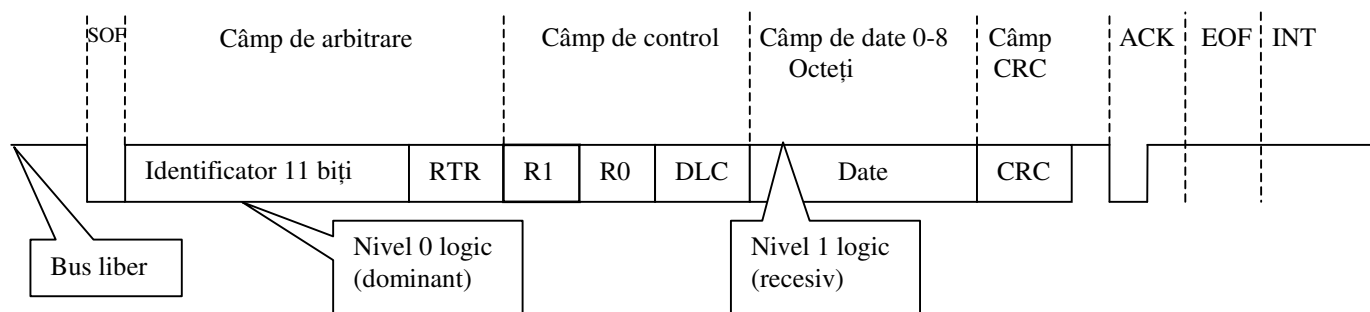


Figura 5.77: Cadru CAN 2.0A

SOF-câmp de start (Start of Frame).

RTR- (Remote Transmission Request) un bit care indică un cadru de cerere de informație. Cadrele de cerere de informații nu au câmp de date.

R0, R1- rezervați

DLC- (Data Length Code) indică numărul de octeți în câmpul de date.

CRC- câmp de 15 biți de verificare a corectitudinii datelor prin cod ciclic. Câmpul CRC se termină cu un bit recesiv.

ACK- este un câmp de confirmare format din 2 biți, primul este un bit dominant trimis de toate nodurile care au primit mesajul iar al doilea bit este recesiv și este un bit de delimitare.

EOF- (End of Frame) este format din 7 biți recesivi

INT- (INTermission) constă în 3 biți recesivi după care magistrala se consideră liberă.

Câmp de arbitrare la cadrul CAN 2.0B

Cadrul de date CAN 2.0B este la fel cu cadrul de date 2.0A cu excepția câmpului de arbitrare care este reprezentat în figura 5.78 și care conține 2 zone, una cu identificatorul pe 11 biți și una cu restul de 18 biți de identificare:

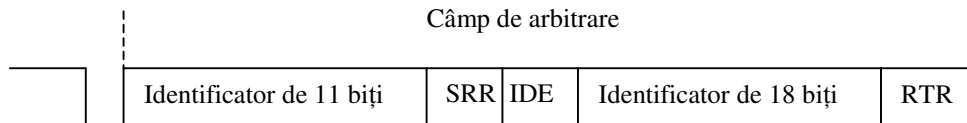


Figura 5.78: Câmp de identificare într-un cadru CAN 2.0B

SRR- (Substitute Remote Request) este un bit recesiv. Dacă se realizează arbitrarea între un cadru 2.0A și unul 2.0B mesajul 2.0A va avea prioritate mai mare dacă identificatorul de 11 biți este identic.

IDE- (Identifier Extension) este un bit care indică dacă cadrul de 2.0A sau 2.0B.

Observații privind compatibilitatea între 2.0A și 2.0B

- Controllerele 2.0B pot transmite sau recepționa mesaje cu ambele structuri.
- Controllerele 2.0A obișnuite, la recepția unui cadru 2.0B vor genera o eroare.
- Există controllere 2.0A (numite pasive) care pot recepționa cadre 2.0B dar le vor ignora.

Cadrul de cerere de informație are structura unui cadru de date cu bitul RTR recesiv și fără câmp de date, figura 5.79:

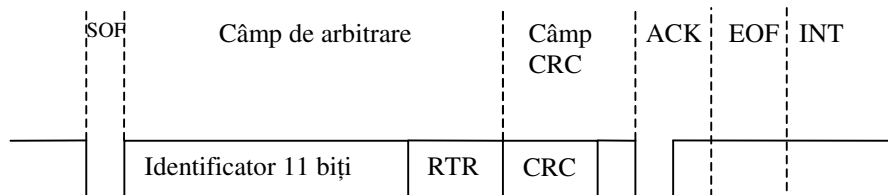


Figura 5.79: Cadrul CAN de cerere de informație

Timpul de bit

CAN este conceput pentru a compensa timpii de întârziere datorate liniilor lungi și a diferențelor dintre tactele fiecărui nod. Tactul CAN dă cuanta de timp T cu care se formează bitul CAN. Fiecare timp de bit este împărțit în 4 sectoare, figura 5.80.

- Segmentul SYNC de sincronizare este intervalul de timp în care se așteaptă frontul activ al impulsului. Are valoarea T .
- Segmentul de propagare compensează întârzierea pe linie și se poate programa între 1 și $8T$.

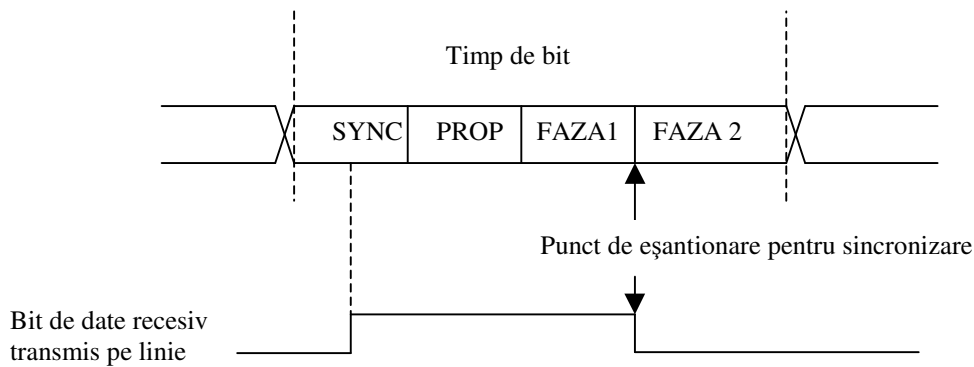


Figura 5.80: Timpul de bit

Segmentele FAZA 1 și FAZA 2 sunt folosite pentru resincronizare. Resincronizarea se face pe frontul de la nivel recesiv la dominant (descrescător). Se definește SJW (Synchronisation Jump Width) ca numărul maxim de cuante T cu care se poate face resincronizarea. Dacă transmițătorul este mai rapid (bit de date mai scurt) frontul descrescător apare în FAZA 1 și FAZA 1 se dublează sau se prelungește cu o cantă T (sau cu mai multe, dar nu mai mult de SJW). Dacă frontul descrescător apare în FAZA 2, FAZA 2 se scurtează. Resincronizarea este necesară pentru recepția corectă a biților recepționați. Codarea la transmisie se face în cod NRZ cu împănare de biți (bit stuffing) și este posibilă recepția mai multor biți cu aceeași valoare logică.

Codarea NRZ

Codarea NRZ asociază bitului de 1 un nivel recesiv și bitului de 0 un nivel dominant. Dacă în șirul de date apar mai mult de 5 valori cu același nivel logic se introduce automat un bit de valoare contrară.

Gestionarea erorilor

Detectarea erorilor și semnalizarea lor face din CAN o magistrală de mare siguranță. Nodurile defecte nu afectează traficul pe magistrală. Procesarea erorilor implică următorii pași:

- Eroarea este detectată de controller (la recepție sau transmisie)
- Se transmite un cadru de eroare
- Se incrementează un registru de eroare
- Mesajul eronat este abandonat în toate nodurile, apoi este retransmis.

Erorile detectate pot fi:

- Erori de bit
 1. Eroare de împănare (bit stuffing). Codarea datelor se face NRZ cu împănare de biți prin adăugarea unui 1 după 5 biți consecutivi de 0 sau un 0 după 5 biți consecutivi de 1. Nodul de recepție detectează un număr mai mare de 5 biți consecutivi cu aceeași valoare logică.
 2. Eroare de bit, când un transmițător care citește tot ce transmite detectează o nepotrivire între ceea ce a transmis și ceea ce a recepționat (în altă parte decât în câmpul de arbitrar sau în câmpul de confirmare ACK) generează o condiție de eroare.
- Erori de mesaj
 1. Eroare de sumă de control CRC
 2. Eroare de cadrare. În cadrul CAN se transmit anumiți biți cu valori predefinite. Dacă receptorul detectează un astfel de bit eronat se generează o eroare (Frame, Format sau Form Error).
 3. Eroare de confirmare, dacă transmițătorul determină că mesajul nu a fost recepționat (nu s-a recepționat ACK).

Un modul CAN poate fi din punctul de vedere al gestionării erorilor în una dintre 3 stări:

- Stare activă, care este modul de operare normală în care apariția unei erori poziționează un bit de eroare
- Stare pasivă, este un mod în care poate opera normal, în condițiile în care nodul a avut probleme dese de transmisie sau recepție a mesajelor.
- Blocat, nu se pot transmite sau recepționa mesaje. Nodul poate ieși din această stare printr-un RESET comandat de calculatorul gazdă.

Numărătorul de erori se modifică în următoarele condiții:

- La o recepție eronată numărătorul rx_count se incrementează cu 1
- La recepția unui bit dominant după transmiterea unui bit de eroare, rx_count se incrementează cu 8.
- Când un transmițător trimite un bit de eroare tx_count se incrementează cu 8.
- După recepția cu succes a unui cadru rx_count se decrementează cu 1.
- Nodurile acceptă maximum 7 biți dominanți după transmiterea unui bit de eroare. Dacă se transmite / recepționează mai mult de 14 biți dominanți sau 8 biți dominanți urmați de un bit de eroare rx_count se incrementează cu 8.

Cadrul de eroare este reprezentat în figura 5.81:

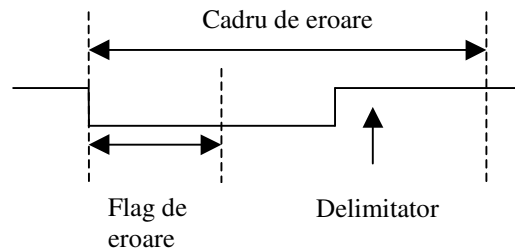


Figura 5.81: Cadrul CAN de eroare

- Flagul de eroare activă conține 6 biți dominanți
- Flagul de eroare pasivă conține 6 biți recesivi, dacă nu sunt suprascrise de biți dominanți de la alte noduri, într-o stare de eroare activă (mai gravă).

Observație: acest cadru nu respectă regula de codare CAN cu bit stuffing pentru că delimitatorul este format din 8 biți recesivi.

Un cadru de eroare este transmis când un nod detectează o eroare. Cadrul transmis nu permite celorlaltor noduri să accepte mesajul eronat. După transmiterea cadrului de eroare toate nodurile detectează violarea de cod și transmit cadre de eroare. După acest pas transmiterea de cadre se reia. Dacă cadrul de eroare este pasiv (6 biți recesivi) magistrala nu se blochează.

Tipuri de CAN

1. Basic CAN echipază controllerele mai ieftine. Conține 2 buffere de recepție de tip FIFO și unul de transmisie. La recepție în timp de MC citește un mesaj, un altul se poate recepționa în FIFO. Mesajele recepționate sunt verificate cu un filtru de acceptanță de 2 octeți pentru a se identifica mesajele care trebuie preluate. Verificarea identificatorilor este hardware și se face pe mai puțini biți pentru a mări viteza. Verificarea finală a identificatorilor se face software. Cadrele de cerere de informație nu sunt generate hardware, ele trebuie generate prin program.
2. Full CAN conține un set de buffere de cadre numite căsuțe poștale. La inițializare fiecărei căsuțe poștale i se asignează un identificator. Cadrele de cerere de informație sunt gestionate hardware. Filtrele de acceptanță verifică în întregime identificatorii și recepția se face în căsuța poștală de destinație.

5.9.2. Interfața CAN

Caracteristicile principale sunt:

- Corespunde specificațiilor CAN 2.0A și 2.0B
- Poate transmite date la recepția unui cadru de cerere de informații
- Dispune de 16 buffere pentru cadre transmise / recepționate
- Filtrele de acceptanță pot compara toți biții sau prin mascare parțială se poate compara un anumit număr de biți
- Se poate obține o rată de transfer între 10kbps și 1Mbps (este nevoie de tact mașină de minimum 8MHz pentru a obține 1Mbps).

Regiștrii CAN pot fi:

- Regiștrii de comandă generală
- Regiștrii de comandă pentru buffere
- Buffere de cadre

Schema bloc simplificată a interfeței CAN este dată în figura 5.82:

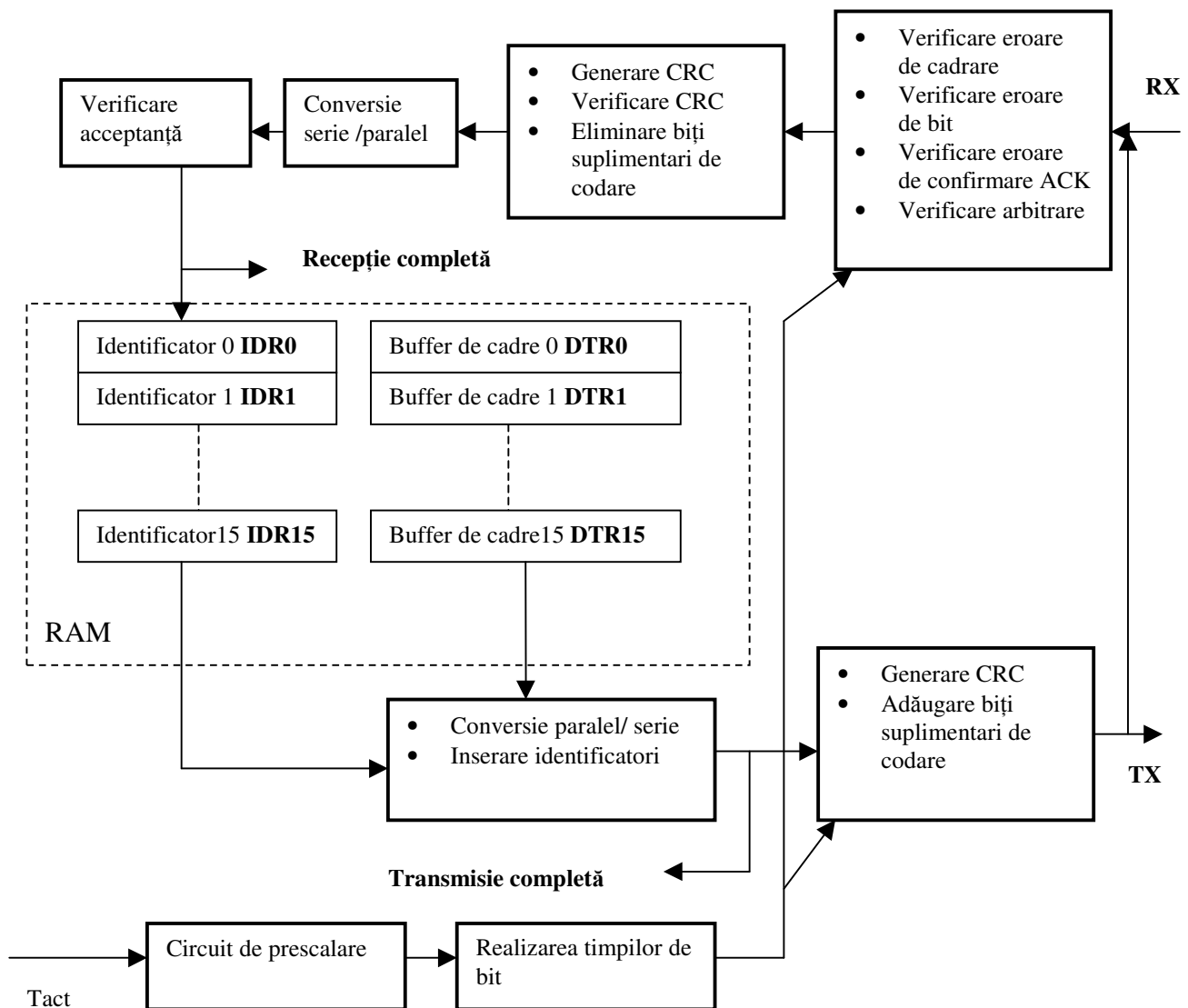


Figura 5.82: Schema bloc a interfeței CAN

Regiștrii CAN de comandă generală

CSR (Control Status Register) (partea Low), un bit stabilește semnificația pinilor (CAN sau pini de I/O de uz general), un bit validează întreruperea la schimbarea stării unui nod, un bit comandă starea de HALT pentru un nod (sau iese din această stare). În CSR (partea High) un bit arată starea transmițătorului (dacă un mesaj este în curs de transmisie), un bit arată că s-a recepționat un mesaj, un bit arată o schimbare în starea nodului iar 2 biți NS1 și NS0 arată starea nodului:

| NS1 | NS0 | Starea nodului |
|-----|-----|-------------------------------------|
| 0 | 0 | stare de eroare activă |
| 0 | 1 | avertizare (stare de eroare activă) |
| 1 | 0 | eroare pasivă |
| 1 | 1 | magistrala blocată |

Starea de avertizare este o stare introdusă în CAN 2.0B, în cazul în care valoarea numărătorului de erori depășește 96.

LEIR (Last Event Indicator Register), bitul RCE (Receive Completion Event) arată că ultimul eveniment a fost o recepție, TCE (Transmit Completion Event) arată că ultimul eveniment a fost o transmisie, NTE (Node Status Transition Event) arată că ultimul eveniment a fost o schimbare a stării nodului. Doar unul dintre biții RCE, TCE sau NTE poate fi activ. Dacă RCE (sau TCE) este activ, 4 biți arată numărul bufferului de cadre (0-15) cu care s-a făcut ultimul transfer.

RTEC (Receive and Transmit Error Counter) arată numărul de erori și poate fi doar citit. Partea Low (8biți) arată numărul de erori la recepție iar partea High (8 biți) arată numărul erorilor de transmisie. Depășirea valorii de 256 la recepție duce la intrarea în starea de eroare pasivă iar depășirea valorii de 256 la transmisie duce la blocarea magistralei.

BTR (Bit Timing Register), 2 biți stabilesc numărul de cuante de timp T pentru FAZA 2, 2 biți stabilesc numărul de cuante de timp T pentru un segment de timp care conține FAZA 1+ timpul de propagare, 2 biți stabilesc numărul de cuante de timp pentru SJW iar 6 biți stabilesc prescalarea tactului CAN (setări specifice tipului de MC folosit).

Regiștrii CAN de comandă pentru bufferele de date

BVALR (Message Buffer Valid Register) arată starea bufferelor de cadre, câte un bit alocat fiecărui buffer. Un bit de 0 arată că bufferul este invalid iar un bit de 1 arată că bufferul este valid.

IDER (Identifier Extension Register) arată pentru fiecare buffer de cadre tipul de identificator folosit, dacă bitul este 0 se folosește formatul standard, iar dacă bitul este 1 se folosește formatul extins pe 29 de biți.

TREQR (Transmission Request Register) arată starea bufferelor de cadre la transmisie, câte un bit TREQ_i pentru fiecare buffer. Un bit de 1 pornește transmisia unui cadru din bufferul corespunzător. Dacă se scrie 1 pentru mai multe buffere transmisia începe cu bufferul cu numărul cel mai mic. Bitul rămâne 1 pe toată perioada transmisiei apoi devine 0.

TRTRR (Transmission RTR Register) conține valoarea biților RTR pentru fiecare buffer de cadre la transmisie. Dacă RTR_i=0 cadrul este de date, iar dacă RTR=1 cadrul este de cerere de informație.

RFWTR (Remote Frame Receiving Wait Register) conține condițiile de start pentru o transmisie, câte un bit pentru fiecare buffer de cadre. Dacă bitul RFWTR_i este 0 transmisia începe imediat. Dacă RFWTR_i este 1 transmisia începe după recepționarea unui cadru de cerere de informație, indicat de bitul RRTR_i din registrul RRTRR. Transmisia începe dacă este comandat un start prin bitul TREQ_i=1 și bitul RTR comandat de TRTRR_i=0 (se indică un cadru de date).

TCANR (Transmission Cancel Register) conține câte un bit pentru fiecare buffer de cadre care anulează o cerere o cerere de transmisie comandată de TREQR.

TCR (Transmission Complete Register) conține câte un bit pentru fiecare buffer de cadre, bit care devine 1 la terminarea transmisiei. Se poate genera o întrerupere dacă aceasta este validată în TIER.

TIER (Transmission Interrupt Enable Register) validează/ invalidează întreruperile generate de terminarea transmisiei cu câte un bit pentru fiecare buffer de cadre.

RCR (Reception Complete Register) conține câte un bit pentru fiecare buffer de cadre, bit care devine 1 la recepția unui cadru. Se poate genera o întrerupere dacă întreruperea este validată. După salvarea mesajului este necesar să se scrie 0 pe poziția acestui bit.

RIER (Reception Interrupt Enable Register) (Transmission Interrupt Enable Register) validează/ invalidează întreruperile generate de terminarea recepției cu câte un bit pentru fiecare buffer de cadre.

RRTRR (Remote Request Receiving Register). După recepția unui cadru de cerere de informație într-un buffer de cadre, bitul RRTR_i corespunzător bufferului respectiv devine 1, deodată cu bitul RC_i din RCR.

RCVRR (Receive Overrun Register). Dacă se recepționează un cadru într-un buffer de recepție *i* și bitul RC_i din RCR este 1 indicând că mesajul anterior nu a fost preluat, ROVR_i devine 1 indicând o eroare de suprascriere.

AMSR (Acceptance Mask Select Register) este masca pentru identificatori (registru de 4 octeți). Fiecare buffer de cadre are alocati 2 biți care semnifică:

| AMS _{i1} | AMS _{i0} | Modul de acceptare al mesajului |
|-------------------|-------------------|---|
| 0 | 0 | comparare pe toți biții identificatorului |
| 0 | 1 | mascare completă |
| 1 | 0 | mască de acceptare în AMR0 |
| 1 | 1 | mască de acceptare în AMR1 |

AMR0/ AMR1 (Acceptance Mask Register 0 /1) sunt regiștrii de 4 octeți care conțin masca pentru identificatori pe 11 biți (AM28-AM18) sau pentru identificatori pe 29 de biți (AM28-AM0).

Buffere de cadre (cadru = mesaj)

Sunt 16 buffere de cadre *i* (*i*=0-15) care conțin câte un registru de identificare IDR_i, un registru DLCR_i care conține numărul de octeți de date din cadru și un registru de date DTR_i. Bufferele sunt folosite la recepție și transmisie. Bufferul cu număr mai mic are prioritate mai mare. La recepție, când un mesaj este acceptat (prin mecanismul de comparare al identificatorului mesajului cu identificatorul stocat în IDR_i) de mai multe buffere, el este stocat în bufferul cu numărul cel mai mic. Dacă se folosește același filtru de acceptanță, în mai multe buffere se pot recepționa mai multe cadre, stocate în ordinea în care au venit, începând cu bufferul cel mai prioritar.

IDR_i este registrul pentru identificatori și conține 4 octeți. În acest registru se pot stoca atât identificatori pe 11 biți cât și pe 29 de biți. Registrul se folosește la recepție indicând identificatorul de acceptanță sau la transmisie prin atașarea acestui identificator mesajului transmis.

DLCR_i stabilește numărul de octeți de date din cadrul de date sau numărul de octeți de date solicitat printr-un cadru de cerere de informație. Cu 4 biți se poate stabili numărul de octeți de la 0 la 8.

DTR_i este un registru de 8 octeți în care se stochează datele care se transmit sau se recepționează începând cu MSB. Dacă numărul de octeți de date se programează cu DLCR_i mai mic de 8, restul de octeți din DTR_i nu contează.

Funcționarea CAN

Transmisia

După scrierea unui 1 pe poziția bitului TREQ_i în registrul TREQR pornește transmisia. TC_i din TCR devine 0. Transmisia pornește efectiv când magistrala este liberă.

Dacă bitul RFWT_i din registrul RFWTR este 0 (care indică faptul că acest cadru nu este un răspuns la o cerere de informație) transmisia începe imediat. Dacă RFWT_i este 1 se așteaptă cadrul de cerere de informație, indicat de bitul RRTR_i din RRTRR.

Dacă o cerere se face către mai multe buffere, transmisia începe cu bufferul cu cel mai mic număr.

Dacă bitul TRTRi din TRTRR este 0 se transmite un cadru de date. Dacă TRTRi este 1 se transmite un cadru de cerere de informație.

Dacă apare o coliziune (eroare de arbitrare) sau apare o eroare la transmisie, bufferul de cadre așteaptă eliberarea magistralei și reîncearcă transmisia.

Transmisia se poate abandona prin scrierea bitului TCANi din TCANR. Transmisia se mai poate abandona dacă în același buffer se recepționează un cadru care a trecut de filtrul de acceptanță.

Dacă transmisia s-a făcut fără erori RRTRi devine 0, TREQi devine 0 și TCi devine 1 și se poate genera o întrerupere dacă este validată de bitul TIEi din TIER.

Recepția

Recepția începe când este detectat un SOF (Start of Frame) pe magistrală.

Identificatorul mesajului recepționat este comparat cu identificatorul stocat în IDRi pe 11 sau 29 de biți, funcție de bitul IDEi din IDER.

Mesajele acceptate sunt stocate în bufferele de cadre (datele în DTRi, lungimea datelor în DLCi și identificatorul în IDRi).

Dacă mesajul este acceptat de mai multe buffere el este stocat în bufferul cu numărul cel mai mic.

Dacă mesajul se stochează într-un buffer de unde nu s-au citit datele (RCi este 1) se semnalizează o eroare de suprascrisere (bitul ROVRi din ROVRR devine 1).

După salvarea mesajului, RCi din RCR devine 1 și se poate cere o întrerupere dacă RIEi din RIER validează întreruperile.

Observații

1. Controllerul CAN integrat în acest tip de MC (MB90350) nu poate recepționa mesaje transmise de el însuși.

2. La MB90350 tactul poate fi modulată, ceea ce nu este admis de CAN. De aceea, în registrul CDMR (CAN Direct Mode Register) trebuie poziționat bitul DIRECT pe 0 dacă modularea este activă și pe 1 dacă modularea nu este activă.

5.9.3. Transceiver CAN Microchip MCP2551

Circuitul transceiver CAN este o interfață între semnalele digitale de la nivelul MC și semnalul diferențial de magistrală CAN. Transceiverul protejează MC de vârfurile de tensiune prezente pe magistrala CAN datorită EMI sau descărcărilor electrostatice. Principalele caracteristici ale circuitului sunt:

- Implementează standardul ISO 11898 pentru nivelul fizic de transmisie
- Viteza maximă de transfer este 1Mbps
- Panta semnalului emis pe magistrală poate fi ajustată extern ceea ce reduce spectrul de frecvență al perturbațiilor generate
- Detectează erorile de masă (permanent dominant) și decuplează driverul de ieșire
- Curentul de funcționare poate fi redus prin intrarea în starea de *standby*
- Are protecție la scurtcircuit, la supratensiune și la supratemperatură
- Se pot conecta până la 112 noduri
- Gama de temperaturi de funcționare poate fi cea industrială (I) -40°C la +85°C sau cea extinsă (E) -40°C la +125°C
- Capsula este PDIP cu 8 pini

Schema bloc a circuitului este dată în figura 5.83:

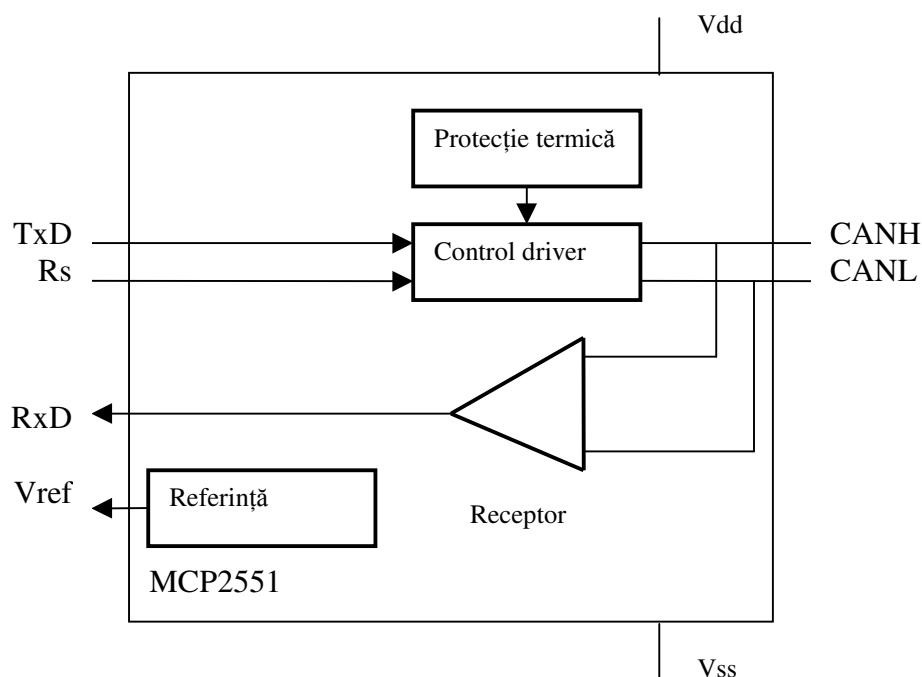


Figura 5.83: Schema bloc a transceiverului CAN MCP2551

Starea logică dominantă este atunci când între CANH și CANL tensiunea diferențială este mai mare decât un prag (1,2V) iar starea recesivă când tensiunea diferențială este mai mică decât un prag (0V).

Circuitul poate fi comandat să intre într-o stare cu consum redus (*standby*) printr-un nivel High pe pinul Rs. În mod standby circuitul citește magistrala CAN (mesaje Low Speed). MC primește datele, supraveghează magistrala și poate comanda viteza nominală, dar primul mesaj poate fi pierdut.

Panta semnalelor emise pe magistrală poate fi controlată cu un rezistor între pinul Rs și masă (control posibil în modul de lucru cu viteză mică). În modul High Speed CAN, pinul Rs se leagă la masă.

Dacă durata ținerii în 0 a magistralei depășește 1,2ms (20 de biți dominanți consecutivi) driverul de ieșire este decuplat de la linie.

Dacă circuitul se supraîncălzește (mai mult de 165°C) circuitul de protecție decuplează driverul de ieșire până la revenirea temperaturii la normal.

Tensiunea de alimentare Vdd este de +5V (între 4,5V și 5,5V) iar tensiunea de referință este Vdd/2.

5.10. Magistrala Flex Ray

Flex Ray este o nouă arhitectură de interfață în care viteza de transfer ajunge la 10Mbps, cu elemente de redundanță pentru a asigura o siguranță mare de utilizare dedicată industriei auto. Arhitectura magistralei este dată în figura 5.84:

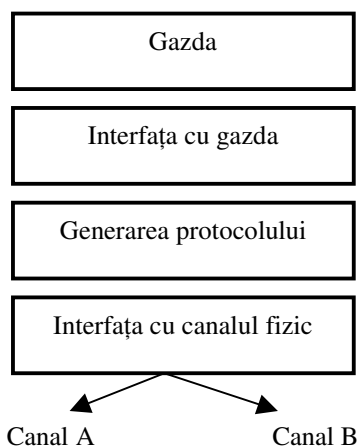


Figura 5.84: Arhitectura magistralei Flex Ray

Topologia rețelei în cazul unui singur canal poate fi de tip magistrală liniară sau stea multiplă iar în cazul a 2 canale cu aceeași configurații de tip magistrală sau stea se pot construi rețele în care canalele sunt redundante. În cazul rețelelor cu 2 canale redundante un canal se poate defecta și sistemul își păstrează funcționalitatea. La nivel fizic interfața admite un gardian de magistrală care realizează detecția erorilor în domeniul timp și care interacționează cu gazda prin comunicarea erorilor, configurare, activare sau dezactivare.

Blocul de generare al protocolului realizează mesajul Flex Ray, detaliat în figura 5.85:

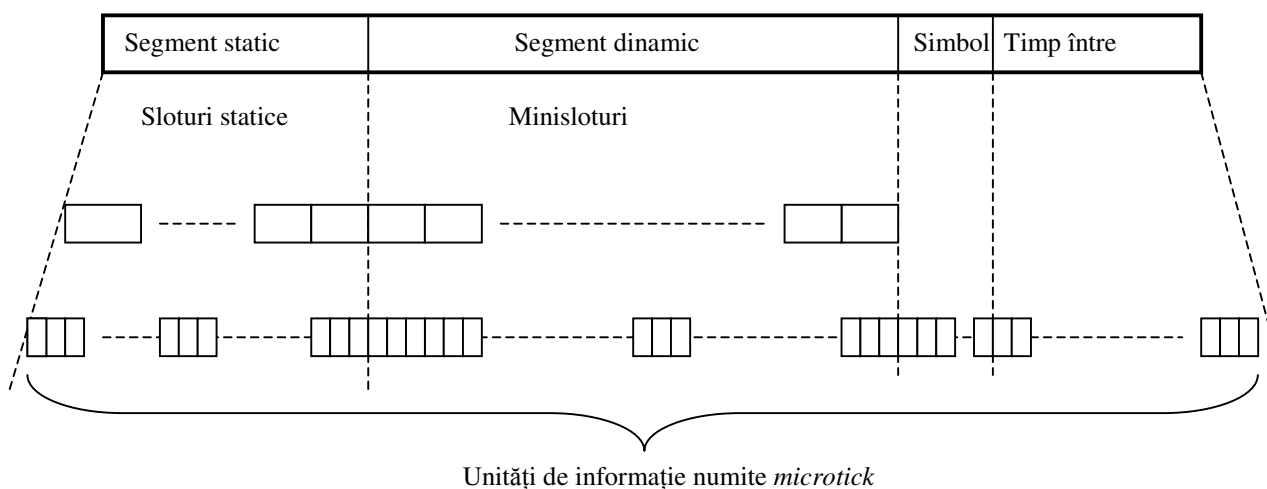
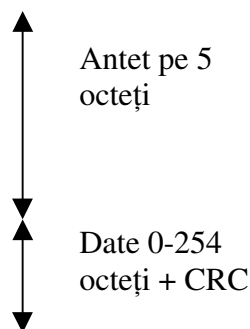


Figura 5.85: Cadrul Flex Ray

Cadrul conține:

- Un bit de gestionare a rețelei
- Un bit care indică un cadru gol
- Un bit de cadrare
- Identificator de cadru pe 12 biți
- Lungimea cadrului (în octeți) codificată pe 7 biți
- CRC pentru antet pe 11 biți
- Identificator de mesaj (opțional) pe 16 biți
- Date, lungime variabilă
- CRC pentru date pe 24 de biți



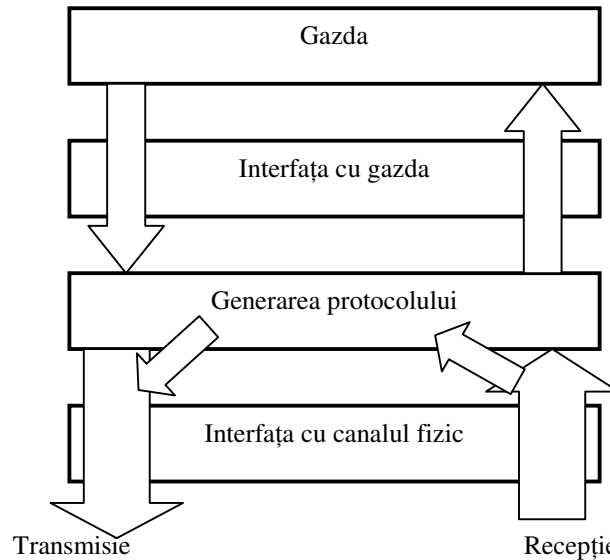
Comunicarea este de 2 tipuri după modul de inițiere:

- Comunicată inițiată la anumite momente de timp în care ciclul de comunicații începe periodic, de exemplu la întreruperile unui timer.
- Comunicare inițiată de un eveniment extern

Mesajele se transmit în cadre. Cadrele pot fi transmise astfel:

- Cadrele statice se transmit repetitiv, de exemplu cadrele de control a funcționării unui subansamblu
- Cadrele dinamice se transmit la cerere, de exemplu informația de diagnostic.

Fluxul de date poate fi urmărit în figura 5.86:



Modulul de atașare / separare protocol separă antetul de date și trimite datele către gazdă la recepție și adaugă informația de protocol la transmisie.

Figura 5.86: Fluxul de date

Structura segmentelor statice transmise în cazul comunicației pe 2 canale este dată în figura 5.87:

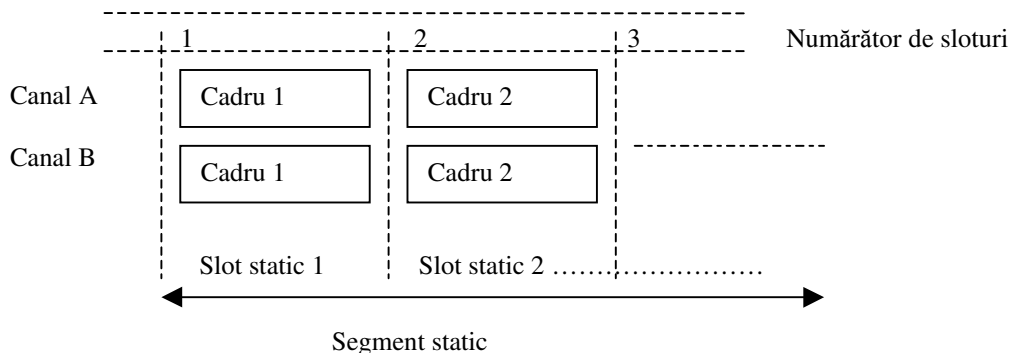


Figura 5.87: Segment static

Lungimea unui slot static este constantă. Un mesaj poate avea un număr variabil de sloturi. Numărătorul de sloturi se incrementează cu 1 după fiecare slot.

La transmisia unui segment dinamic alocarea timpului pentru cele 2 canale este dinamică, figura 5.88:

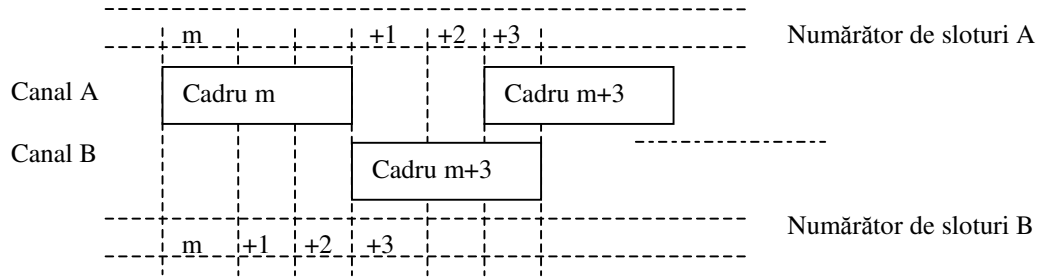


Figura 5.88: Segment dinamic

Sincronizarea se realizează de către generatorul de protocol la nivel de unitate de informație (microtick). Cea mai mare deviație permisă între tactele nodurilor conectate este de o microsecundă. Sincronizarea se execută cu ajutorul unor cadre de sincronizare trimise de noduri.

Serviciul de gestionare al erorilor se bazează pe principiul de a nu renunța niciodată la transferul unui mesaj. O eroare gravă este de exemplu pierderea sincronizării (prea puține cadre de sincronizare recepționate de la un nod).

Serviciul de simboluri contribuie la mărirea siguranței în funcționare. Simbolurile sunt trimise la sfârșitul cadrului de către controller și trebuie recepționate de către controllerul destinație formând un sistem de supraveghere de tip Watchdog. Simbolurile pot fi de stare normală sau de alarmă.

Serviciul de economie de energie este folosit pentru micșorarea energiei consumate. Ieșirea din starea de adormit se poate face la cererea oricărui nod care trimite un cadru de "trezire".

Serviciul de diagnostic conține 2 tipuri de teste:

- Monitorizarea cu gardianul de magistrală, executată de generatorul de protocol
- Teste de comunicație solicitate de gazdă.

Rezultatele testelor sunt comunicate gazdei care poate decide excluderea nodului din comunicație.

În cazul în care comunicația este inițiată de un eveniment extern (Event Triggered Mode) cadrul curent este abandonat, figura 5.89:

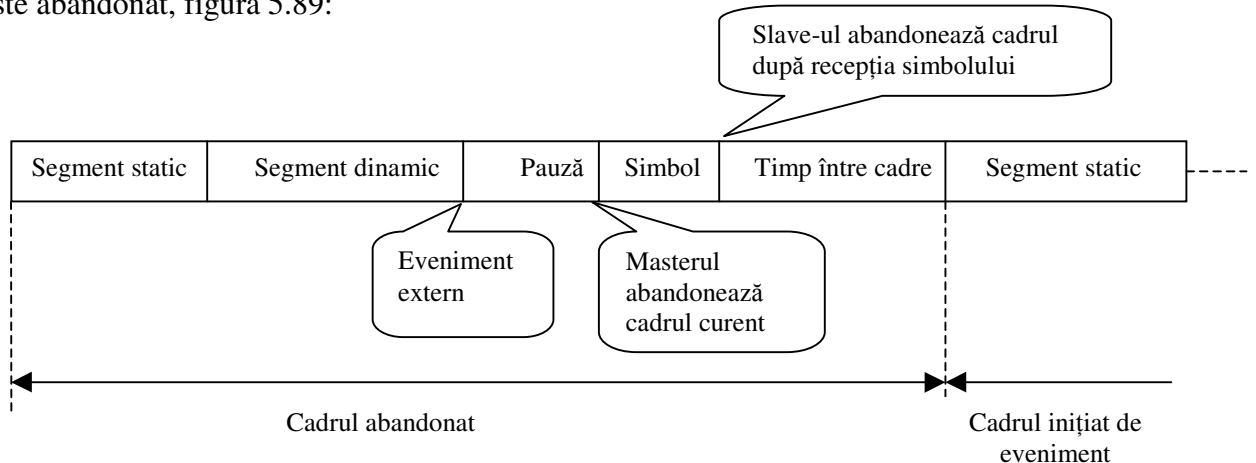


Figura 5.89: Cadrul inițiat de un eveniment extern