

5.INTERFEȚE INTEGRATE

Interfața integrată este modulul electronic integrat în microcontroller care face legătura între unitatea centrală și procesul extern. O schemă bloc simplificată este dată în figura 5.1:

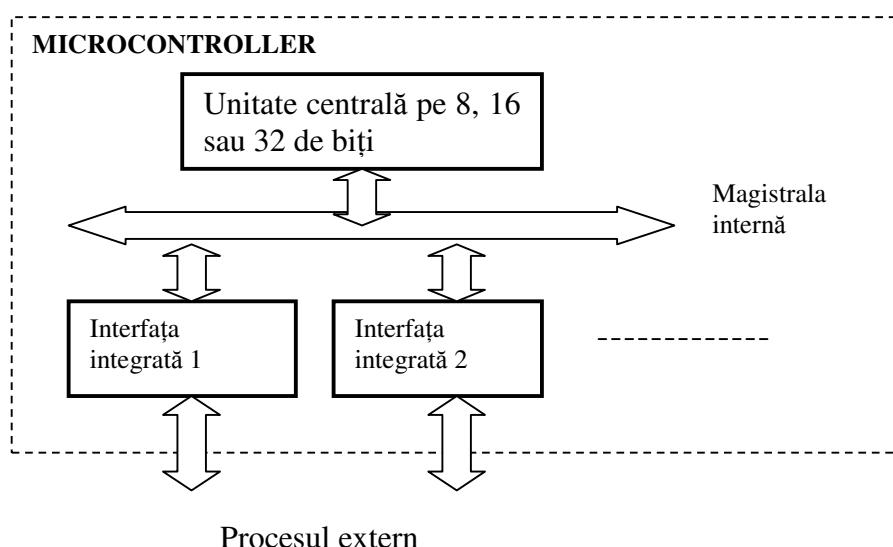


Figura 5.1: locul interfeței între unitatea centrală și procesul extern

Procesul extern poate fi:

- Un proces analogic și interfețele sunt în acest caz convertoare A/D sau D/A
- Un proces digital, de exemplu comutatoare, relee, alte circuite integrate digitale, calculator PC etc.

După modul de transfer al datelor interfețele integrate pot fi:

- Paralele, de exemplu porturile de I/O de uz general, interfața cu afișajul LCD etc.
- Seriale, de exemplu UART, CAN, I²C etc.

Piniile interfețelor integrate pot avea de regulă semnificații multiple, duble sau triple. Piniile pot fi definiți ca pini de I/O de uz general sau pot primi o semnificație particulară.

În evoluția MC și în general a tehnicii de calcul se constată o tendință către utilizarea interfețelor seriale mai mult decât a celor paralele. Astfel de exemple sunt la tot pasul, apariția hard discurilor Serial ATA și răspândirea lor în dauna celor IDE, apariția magistralei USB și înlocuirea interfeței paralele CENTRONICS pentru imprimante etc. Din schema bloc a MC se poate remarca că el este dotat cu mult mai multe interfețe seriale decât paralele. Chiar și la interfețele seriale se remarcă scăderea numărului de fire de transmisie pentru scăderea costurilor, și vor fi prezentate în această idee transmisii LIN și CAN pe un singur fir.

O schemă bloc a unei interfețe cu pini cu semnificație dublă este dată în figura 5.2:

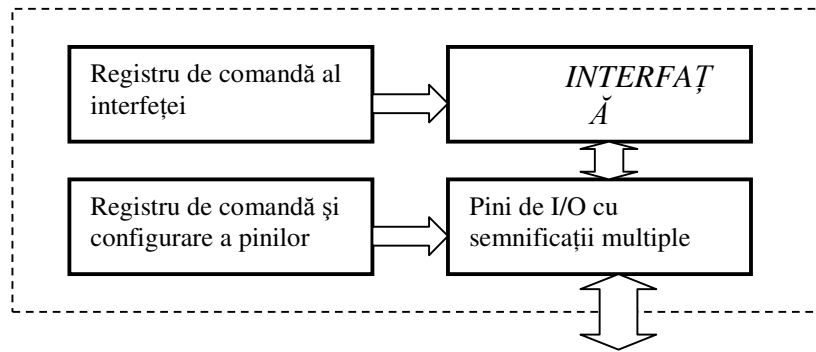


Figura 5.2: schema bloc a unei interfețe cu pini cu semnificații multiple

Registru de comandă al interfeței conține un bit care definește pentru pinul de I/O semnificația de pin de uz general sau pin specific interfeței. De exemplu la UART bitul de comandă se află în registrul SMR (Serial Mode Register), la PPG bitul se află în registrul PPGC (PPG Operation Mode Control Register) etc.

Protocoale de comunicație

Aceste protocoale definesc setul de reguli la schimbul de informație între 2 circuite conectate informațional. Rolul protocoalelor este de a sincroniza transferul între 2 circuite în care procesele au loc cu viteze diferite (cu tact diferit). Protocolul de comunicație este specific fiecărui tip de interfață și va fi analizat în acest capitol pentru fiecare tip de interfață prezentată.

Protocoalele se clasifică în:

- Protocoale realizate cu linii de semnal dedicate
- Protocoale realizate cu informația în mesaj, care pot fi:
 1. Informația suplimentară de sincronizare (informația de cadrare) este formată din 1 sau 2 biți care încadrează mesajul (UART)
 2. Informația suplimentară este formată din cadre de semnalizare (USB, TCP-IP etc.)

Transferuri avansate

Un mod avansat de transfer permite ca transferul să înceapă în momente de timp arbitrare, la cererea unei întreruperi de către evenimentul (procesul) extern sau la îndeplinirea unei condiții de întrerupere la interfața internă.

Schema bloc a modulelor care participă la un transfer avansat este dată în figura 5.3:

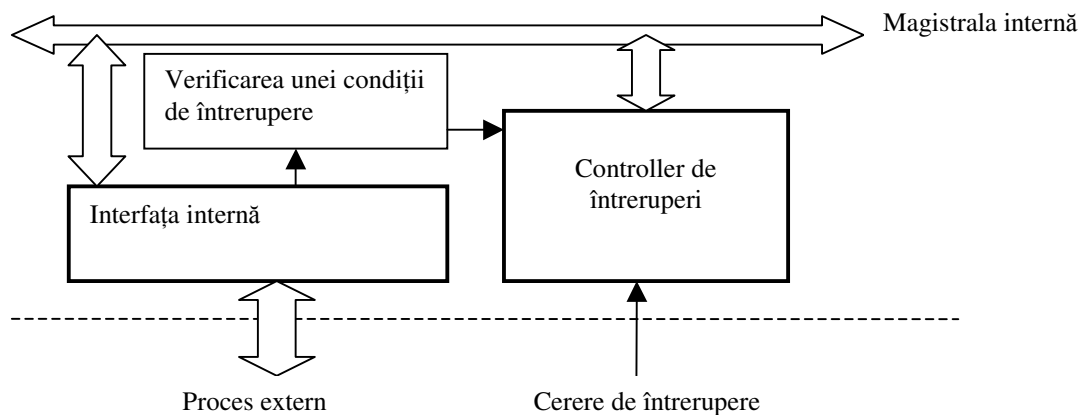


Figura 5.3: schema bloc a unui transfer avansat

Procesul extern poate cere direct o întrerupere la care se poate programa polaritatea activă (front crescător, descrescător, ambele sau nivel) care este analizată din punct de vedere al priorității în controllerul de întreruperi și este transmisă unității centrale. După servirea întreruperii trebuie să se reseteze flagul de cerere de întrerupere poziționat de cererea de întrerupere.

Lucrul în întreruperi poate fi declanșat și de o cerere generată de o interfață internă la verificarea unei condiții, de exemplu emisia sau recepția unui caracter, trecerea prin zero a unui numărător (prin urmare marcarea unui interval de timp) etc. Cererea poziționează un flag de cerere de întrerupere care trebuie resetat după servirea întreruperii. Cererea este analizată de controllerul de întreruperi și transmisă unității centrale.

Transferul propriu zis de date are loc în rutina de servire a întreruperii prin intermediul acumulatorului sau prin serviciul EI²OS respectiv DMA și atunci datele nu mai trec prin acumulator, crescând astfel viteza transferului.

5.1. Interfețe seriale UART

5.1.1 Interfața serială sincronă / asincronă LIN UART

LIN UART este un modul de comunicații seriale sincrone sau asincrone cu dispozitive externe având și posibilitatea de a lucra în standardul LIN (Local Interconnect Network). Comunicația poate fi bidirecțională și se poate lucra în modul *master/slave*.

Funcțiile și caracteristicile UART sunt:

- Mod de transmisie *full duplex*
- Intrarea serială este eșantionată de 5 ori în timpul unei celule bit
- Transferul poate fi sincron sau asincron
- Rata de transfer este programată cu un generator de rată de Baud dedicat care constă într-un numărător de 15 biți reîncărcabil
- Cuvintele de date pot fi pe 7 sau 8 biți. În mod sincron și LIN cuvintele pot fi doar de 8 biți.
- Este posibilă funcționarea în întreruperi și se pot folosi următoarele cereri de întrerupere:
 - la recepție – recepție completă, eroare de recepție, detecție de *break* (pauză în transmisie prin trecerea liniei în stare *space*)
 - la transmisie – registru de transmisie gol
 - detecția unui câmp de sincronizare LIN (LSYN)
- Funcționarea *master slave* permite operarea *master*-ului cu mai multe circuite *slave*
- Ceasul serial poate fi conectat permanent la SCK, astfel încât un transfer sincron poate fi executat cu bit de *start* și *stop*.

Modulul LIN UART are 4 moduri de funcționare determinate de biții de mod MD0 și MD1 din registrul SMR (Serial Mode Register), conform cu tabelul următor:

Moduri de operare	Lungime cuvânt	Sincronizare	Biți de stop
0 (asincron normal)	7 sau 8 +paritate	asincron	1 sau 2
1 (asincron multiprocesor)	7 sau 8 fără paritate +1 bit de selecție adrese /date	asincron	1 sau 2
2 (sincron normal)	8+paritate	sincron	0 sau 1 sau 2
3 (LIN)	8	asincron	1

În modul 0 LIN UART lucrează în mod bidirecțional serial asincron, în modurile 2 și 3 în mod bidirecțional serial ca *master* sau *slave* (sincron respectiv asincron) iar în modul 1 bidirecțional asincron ca *master* sau *slave* în comunicații multiprocesor.

Datele sunt codificate NRZ, adică șirul de date 0110100 se va codifica astfel, figura 5.4:

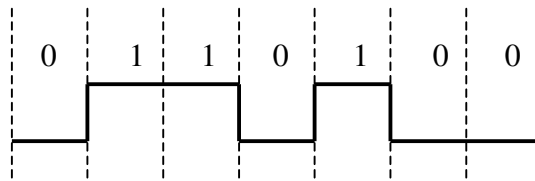


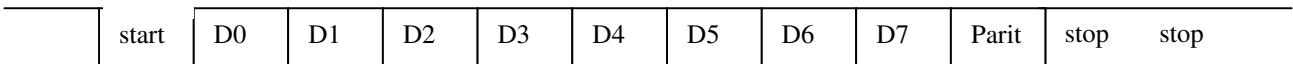
Figura 5.4: șir de date codate NRZ

NRZ nu este o codare autosincronizabilă.

Operarea în mod asincron (0 și 1)

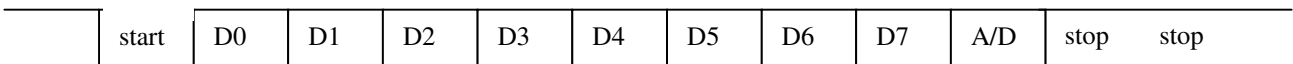
Acest mod de operare are loc în condițiile în care tactul nu se transmite pe linie, caracterele se transmit asincron dar biții în caracter se transmit cu un tact standard, care poate fi ales la transmițător și la receptor. Tactul de transmisie se numește rata de Baud și trebuie să fie același la transmițător și receptor.

Transferul începe cu un bit de *start* (nivel 0) și se termină cu unul sau 2 biți de *stop*. Datele sunt transmise începând cu LSB sau MSP, programabil. Bitul de paritate este opțional și este transmis între ultimul bit de date și primul bit de *stop*. Lungimea cuvintelor de date poate fi de 7 sau 8 biți, cu paritate sau fără, 1 sau 2 biți de *stop*. Structura cuvintelor este dată în figura 5.5:



Biții D7, paritate și un bit de stop pot să lipsească.

Mod 0



Biții D7, și un bit de stop pot să lipsească. Bitul A/D stabilește dacă se transmite o adresă sau o dată

Mod 1

Figura 5.5: structura cuvintelor în mod 0 și 1

În modul 1 nu se poate lucra cu paritate.

Conectarea circuitelor în mod 0 și 1 bidirecțional este dată în figurile 5.6 și 5.7:

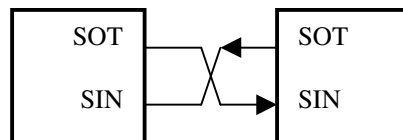


Figura 5.6: conectarea a 2 circuite în mod 0 bidirecțional

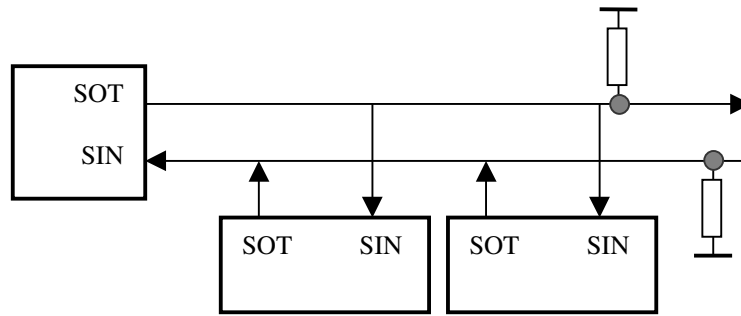
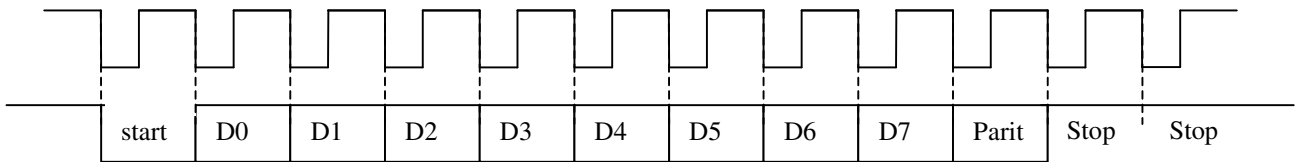


Figura 5.7: conectarea circuitelor în mod 1 bidirecțional multiprocesor

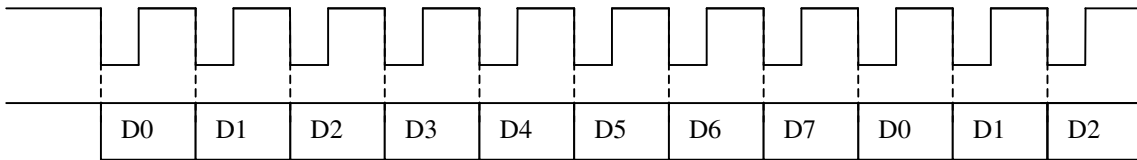
Operare în mod sincron (2)

Transmisia în mod sincron are loc în prezența unui tact transmis o dată cu datele. Datele pot fi transmise fără separatori între octeți și în acest caz tactul este transmis numai când se transmit date sau în același format ca la transferul asincron (bit de *start*, *stop* și paritate), caz în care tactul se transmite permanent, figura 5.8:



Date cu separare

Un bit de stop poate să lipsească.
Tactul este prezent permanent.



Date fără separare

Tactul este generat doar când se transmit date, altfel linia este în stare *mark*.

Figura 5.8: operarea în mod sincron

Tactul de transmisie poate fi activ pe front descrescător (ca în figură) dar se poate programa transmisia cu tact inversat, frontul activ fiind cel crescător. Transmisia poate fi realizată și cu tact întârziat, astfel, figura 5.9:

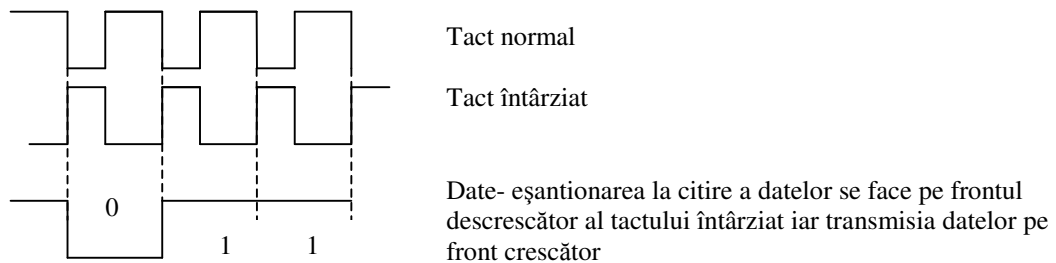


Figura 5.9: transmisia cu tact întârziat

Conectarea circuitelor în mod 2 serial sincron bidirecțional este dată în figura 5.10:

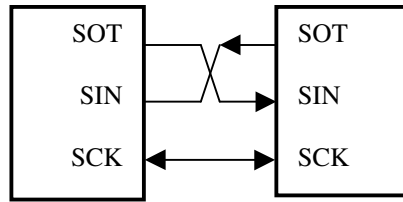


Figura 5.10: conectarea a 2 circuite în mod sincron

Operare în mod LIN (3)

LIN UART poate lucra în acest mod ca *master* sau *slave*. Formatul datelor este pe 8 biți fără paritate, un bit de *stop*, cel mai puțin semnificativ bit se transmite primul. Viteza de transmisie este stabilită de *master* iar circuitele *slave* trebuie să se sincronizeze la această viteză. *Master*-ul generează pe pinul serial de ieșire un nivel *space* pe perioada 13-16 biți, ceea ce constituie semnal de sincronizare pentru slave și începutul unui mesaj LIN. După aceasta se transmite un octet de 55H.

Ca și *slave*, LIN UART recepționează trecerea liniei în *space* și trebuie să calculeze rata de Baud din cuvântul 55H. Aceasta se realizează cu circuitul de captură care este legat intern la LIN UART și care măsoară timpul de recepție pentru octetul 55H. A opta parte a acestui timp este timpul de bit.

Conectarea circuitelor în mod LIN se face cu circuite *transceiver* LIN, ca în figura 5.11:

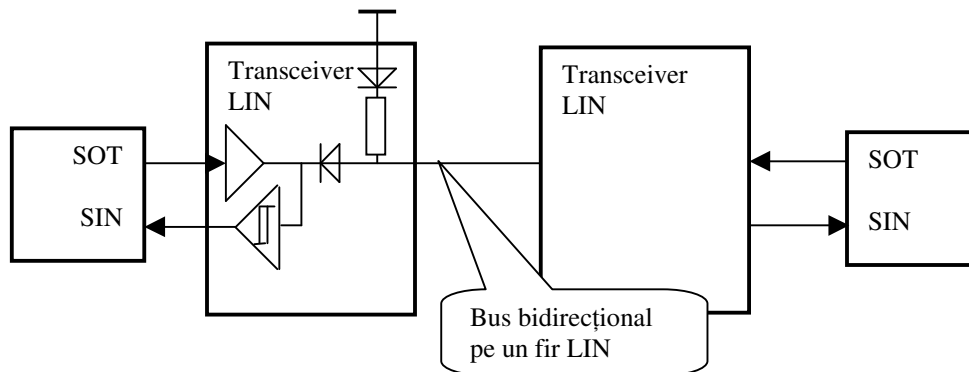


Figura 5.11: conectarea a 2 circuite în mod LIN cu transceivere

Regiștrii LIN UART

SCR (Serial Control Register) conține: validare transmisie, validare recepție, ștergerea indicatorilor de eroare la recepție, bit de adresă sau date (în mod 1), lungime caracter (7 sau 8 biți), biți de *stop* (1 sau 2), validare paritate, paritate pară sau impară.

SMR (Serial Mode Register) selectează modul de operare astfel: stabilește dacă pinii sunt I/O de uz general sau LIN UART, un bit generează un Reset soft, selectează ca și tact de transmisie tactul dat de generatorul de rată de Baud, un ceas extern sau un ceas extern divizat de generatorul de rată de Baud, iar 2 biți stabilesc modul de lucru 0, 1, 2 sau 3.

SSR (Serial Status Register) validează /invalidază întreruperile generate la transmisie /recepție, stabilește dacă primul bit trimis este MSB sau LSB și conține următorii biți de stare: registru de transmisie gol, registru de recepție plin, eroare de cadrare (dacă nu s-a recepționat numărul stabilit de biți de *stop*), eroare de overrun la recepție (dacă s-a recepționat un caracter înainte ca cel precedent să fie preluat), eroare de paritate.

RDR/TDR (Reception and Transmission Data Register) conțin datele recepționate respectiv datele de transmis.

ESCR (Extended Status/Control Register) (folosit în mod 2 și 3) stabilește frontul crescător sau descrescător pe care se face eșantionarea datelor în mod 2, stabilește dacă tactul se transmite continuu, 2 biți stabilesc durata pauzei LIN (13,14,15 sau 16 biți), se validează întreruperile la detecția unei pauze LIN, un bit poate seta direct ieșirea serială SOT și un bit validează această posibilitate.

ECCR (Extended Communication Control Register) (folosit în mod 2 și 3), un *flag* indică starea transmisiei (se transmit date sau nu), un *flag* indică starea recepției, se stabilește dacă în mod 2 se inserează biți de *start* sau *stop*, validează întârzierea tactului, stabilește funcționarea în mod 2 ca *master* sau *slave*, comandă generarea unei pauze LIN.

BGR0 și 1 (Baud Rate Reload Counter Register 0 și 1) asigură divizarea tactului entru transmisia serială asincronă. De exemplu, la rata maximă de Baud de 4MBd, la un tact de 24MHz factorul de divizare este 5, rezultând o deviație de 0. La rata de 115200, la același tact factorul de divizare este 207 cu o deviație de 0,16%. Aceste date se pot calcula sau prelua din tabele.

Generatorul de rată de Baud

Se pot selecta următoarele tacte de transmisie/ recepție:

- Tact stabilit de generatorul de rată de Baud, care este un numărător de 15 biți cu reîncărcare, valoarea de reîncărcare fiind stabilită de regiștrii BGR0 și 1.
- Tact extern furnizat prin pinul SCK (la modul sincron). Sunt posibile frecvențe mai mici decât tactul MC divizat cu 4 (6MHz la un tact de 24MHz).
- Tact extern furnizat generatorului de rată de Baud (în loc de tact intern). Acest mod de lucru a fost conceput pentru a putea conecta cristale specifice transferului serial asincron și de a face divizarea programabilă a acestor frecvențe.

Schema bloc a generatorului de rată de Baud, figura 5.12:

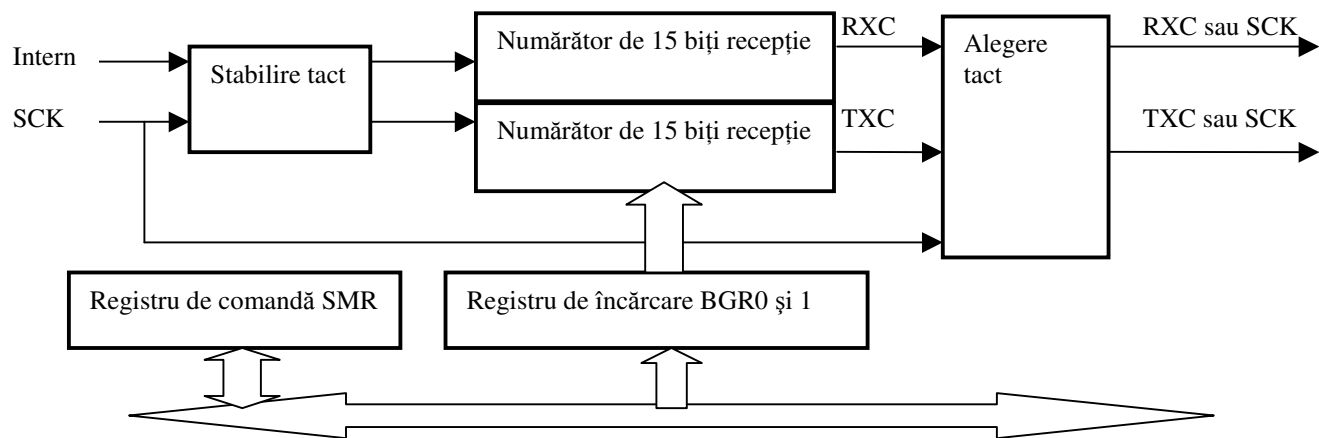


Figura 5.12: schema bloc a generatorului de rată de Baud

Numărătoarele se încarcă cu valoarea din BGR0 și 1 și numără până la semnalizarea depășirii, când se reîncarcă. La fiecare asemenea ciclu se generează RXC respectiv TXC. Registrul SMR programează selecția tactului și modul de lucru sincron/ asincron. Regiștrii sunt conectați la magistrala internă a MC.

Schema bloc a LIN UART și circulația datelor

O schemă simplificată a modului LIN UART este dată în figura 5.13:

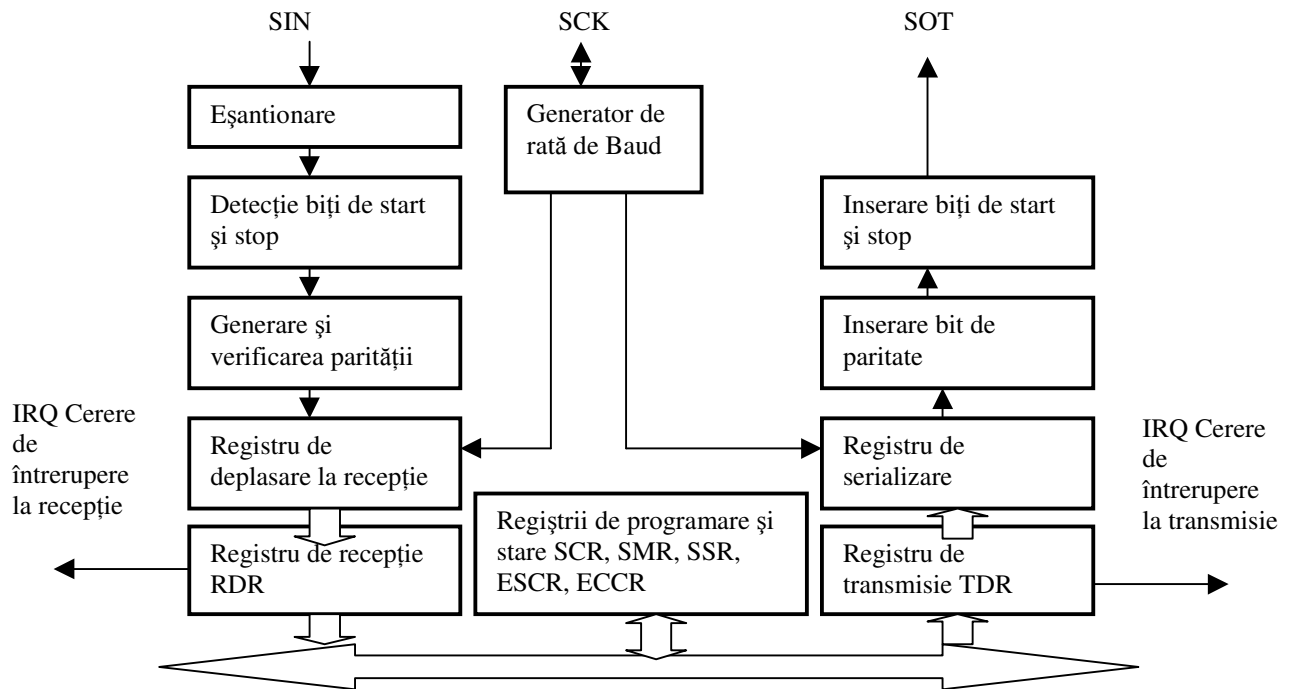


Figura 5.13: schema bloc a modului LIN UART

Sunt reprezentate calea de recepție în stânga și calea de transmisie în dreapta. Datele seriale de intrare sunt eșantionate, se detectează biții de start și stop și când este recepționat un cuvânt se verifică paritatea (dacă s-a programat transfer cu paritate). Cuvântul recepționat este transformat în format paralel cu tactul dat de generatorul de rată de Baud. Când s-a terminat recepția unui caracter se poate cere o întrerupere de recepție. Traseul este invers la transmisie, se serializează caracterul și se înserează biții de paritate, start și stop. Cu regiștrii de programare și stare se asigură funcționarea dorită a modului LIN UART.

Comunicarea în mod 0 sau 2 are loc astfel:

- Se programează LIN UART
- La transmisie se trimite un cuvânt în registrul de transmisie (la o întrerupere de registru gol)
- La recepție se citește registrul de recepție (la o întrerupere de registru de recepție plin sau prin testarea unui *flag*).

În mod 1 *master*-ul trimite un cuvânt care conține adresa *slave*-ului, ceea ce este semnalizat de bitul A/D (ultimul bit înainte de *stop*). Fiecare *slave* verifică adresa, dacă mesajul îi este adresat. Transferul are loc astfel:

- Se programează LIN UART
- La transmisie se trimite un cuvânt în registrul de transmisie cu adresa (bitul A/D=1)
- Se trimite adresa *slave*-ului
- Se trimite un cuvânt cu A/D=0
- Se trimite cuvântul de date în registrul de transmisie
- La recepție se verifică bitul A/D
- Dacă A/D=1 se verifică adresa
- Dacă adresa coincide se verifică bitul A/D
- Dacă A/D este 0 se recepționează date prin citirea registrului de recepție.

5.1.2.Particularități ale canalelor seriale la alte familii

Particularități ale canalelor seriale la familia MB95xxxx

Toate MC din familie au 2 tipuri de module seriale:

- **1.LIN UART** are structura și funcționarea identică cu cea prezentată. Rata maximă de transfer este de 2MBd.

- **2.UART/SIO** care admite următoarele moduri de operare:

-mod 0 asincron, 5-8 biți/cuvânt, lucru cu paritate sau fără, 1 sau 2 biți de stop

-mod 1 sincron, 5-8 biți/cuvânt

Regiștrii folosiți:

SMC1 și **SMC2** (Serial Mode ontrol Register) programează modul de operare, selectează dacă primul bit transmis este LSB sau MSB, selectează modul de operare, lungimea cuvântului și paritatea.

SSR (Serial Status Register) indică starea circuitului și erorile de recepție.

RDR/ TDR sunt regiștrii de date recepționate respectiv transmise.

Modulul UART SIO conține un generator de rată de Baud dedicat. Regiștrii care comandă generatorul sunt:

PSSR (UART/SIO Prescaler Selecton Register) în care cu 2 biți se poate seta divizarea cu 1, 2, 4 sau 8 iar un bit validează funcționarea generatorului.

BRSR (UART/SIO Baud Rate Setting Register) de 8 biți care încarcă valoarea de divizare în generatorul de rată de Baud. Acest registru se poate încărca când generatorul este oprit. Cei 2 regiștrii de divizare dau în total o divizare posibilă pe 10 biți. De exemplu la un tact de 10MHz, dacă rata totală de divizare este de 20 (00H în PSSR și 20D în BRSR) rata de Baud va fi 125000 (rata de transfer maximă). În UART tactul se divizează de 4 ori.

Particularități ale canalelor seriale la familia MB91360

MC pe 32 de biți din familia MB91360 conțin 3 tipuri de interfețe seriale:

- **1.LIN UART** are structura și funcționarea identică cu cea prezentată. Rata maximă de transfer este de 2MBd.

- **2.SIO** este o interfață serială sincronă care poate lucra în 2 moduri:

-cu tact intern

-cu tact extern (pinul SCK)

Regiștrii:

SMCS (Serial Mode Control Status Register) stabilește dacă primul bit transmis este LSB sau MSB, selectează tactul, validează întreruperile, un bit pornește transferul și un bit forțează oprirea transferului.

SDR (Serial Shift Data Register) conține datele care se transmit sau care se recepționează.

CDCR (Serial I/O Prescaler) cu 4 biți poate diviza tactul intern pentru a obține diferite rate de transfer.

- **3.UART** este o interfață serială asincronă *full duplex* care suportă mod multiprocesor.

Tactul poate fi furnizat de un timer intern (numit U-Timer) sau tact extern (neimplementat la MB91360). Ca mod de transfer suportă și modul DMA.

Regiștrii:

SMR (Serial Mode Register) programează modul de operare (mod normal sau multiprocesor) și tactul (U-Timer sau extern)

SCR (Serial Control Register) validează paritatea și selectează paritate pară sau impară, 1 sau 2 biți de *stop*, 7 sau 8 biți de date, stabilește bitul A/D pentru modul multiprocesor, anulează erorile de recepție, validează recepția și validează transmisia.

SIDR/SODR (Serial Input/ Output Register) sunt regiștrii de date recepționate respectiv transmise.

SSR (Serial Status Register) indică o eroare de paritate, de depășire (*overrun*) de cadrare, indică registrul de recepție plin, registrul de transmisie gol, validează lucrul cu întreruperi la recepție, respectiv transmisie.

ULS (UART Level Select Register) poate comanda inversarea intrării/ ieșirii seriale, validează tactul pentru U-Timer și validează UART.

Modulul U-Timer este un timer pe 16 biți folosit pentru generarea ratei de Baud. Timerul este controlat de următorii regiștrii:

UTIM (U-Timer Register) este un numărător reîncărcabil

UTIMR (U-Timer Reload Register) stochează valoarea cu care se reîncarcă UTIM după ce ajunge la 0.

UTIMC (U-Timer Control Register) validează timerul, un bit comandă resetarea UTIM, un bit indică faptul că UTIM a ajuns la 0.

Numărătorul UTIM numără în jos tactul intern sau un tact extern până la atingerea valorii 0 când este reîncărcat cu valoarea din UTIMR. La fiecare trecere prin 0 este dat un impuls de tact pentru UART.

Observație: tipurile mai vechi de MC pe 32 de biți (de exemplu MB911xx) conțin doar canale seriale de tip UART.

5.2. Convertoare analog digitale și digital analogice

5.2.1. Convertorul AD

Ca și modul de conversie analog digital de referință este descris convertorul din MB90350. Acest convertor are următoarele caracteristici:

- Timp de conversie 1,9μs minimum, pentru un canal (la un tact de 24MHz)
- Principiu de conversie prin aproximări succesive cu circuit de eșantionare și menținere
- Rezoluția 8 sau 10 biți, programabilă
- 1-24 de canale de conversie, selectabile soft. Ca mod de conversie se poate selecta ca un singur canal să fie convertit (conversie simplă) sau un grup de canale să fie convertite pe rând automat (conversie multiplă)
 - Modul de conversie poate fi:
 - Individual, fiecare canal poate fi convertit la o comandă
 - Continuu, startul unei conversii are loc automat după terminarea conversiei anterioare
 - Stop, are loc o conversie, apoi convertorul se oprește (intrând în starea de Stop) și așteaptă o nouă activare
 - conversie poate fi pornită astfel:
 - Printr-o comandă software
 - De la un timer
 - Cerere externă de la un pin extern (pe front descrescător)

Schema bloc tipică a unui convertor cu aproximări succesive este dată în figura 5.14:

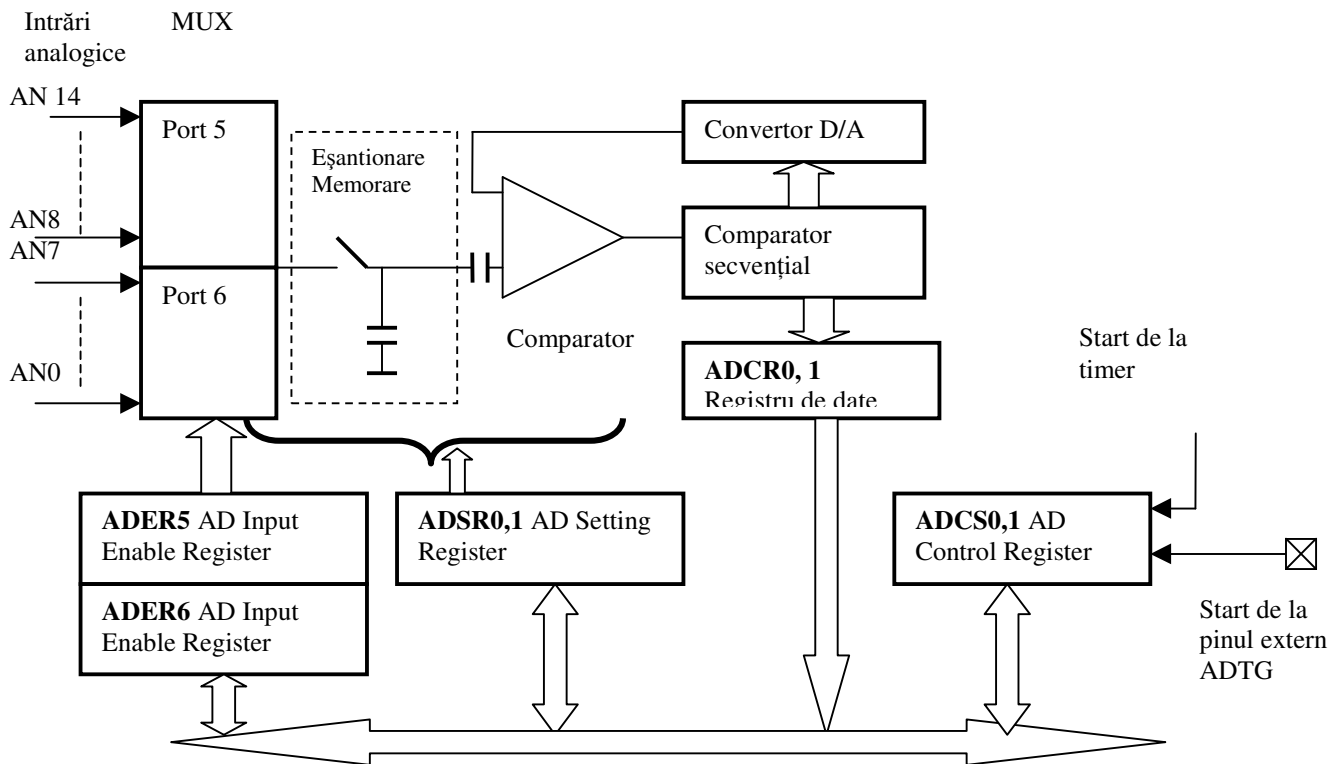


Figura 5.14: schema bloc a convertorului A/D

Semnalul analogic selectat de multiplexorul analogic comandat de regiștrii ADER5 și 6 este eșantionat și memorat pe timpul conversiei. Convertorul prin aproximări succesive este tipic și conține un convertor D/A și un comparator. Rezultatul conversiei este reținut în regiștrii ADCR0 și 1. Regiștrii ADSR 0 și 1 programează modurile de lucru pentru multiplexor, comparator și circuitul de eșantionare memorare.

Regiștrii

ADER5, 6 (AD Input Enable Register) cu câte un bit (în H) selectează unul dintre canalele de intrare analogice. Cu nivel L se validează funcționarea liniilor ca linii de I/O de uz general.

!Observație

După RESET semnificația liniilor este de intrări analogice.

ADCS0 (AD Control Status Register) stabilește cu 2 biți modul de lucru (individual, continuu sau stop), un bit stabilește rezoluția (8 sau 10 biți)

ADCS1 (AD Control Status Register) conține un flag pentru a indica că AD este ocupat cu o conversie (BUSY), un flag indică o cerere de întrerupere (când datele s-au scris în registrul de date), un bit comandă validarea întreruperilor, un flag indică faptul că AD este oprit (PAUSE), 2 biți stabilesc modul de start (prin soft, de la un pin extern sau soft, timer sau soft, toate variantele), un bit porniște conversia (Start).

Observație

Un singur registru de date stochează rezultatul conversiei (ADCR). În conversia continuă datele rezultate din conversie se suprascriu în acest registru și distrug datele anterioare. Pentru a preveni acest lucru AD se oprește (PAUSE) după o conversie până când datele sunt preluate, lucru indicat de trecerea lui INT în 0. Trecerea lui INT în 1 se face când s-a terminat o conversie și rezultatul este în registrul de date. Trecerea în 0 a lui INT nu se face automat, este nevoie sarcina rutinei de întrerupere să facă acest bit 0 după ce a preluat și prelucrat datele.

ADCR0, 1 (AD Data Register), ADCR0 stochează 8 biți mai puțin semnificativi, ADCR1 2 biți mai semnificativi ai conversiei (pe poziția D9, D8, D15 fiind cel mai semnificativ bit). Regiștrii de date sunt scriși continuu la terminarea unei conversii. În modul cu rezoluția de 8 biți rezultatul este în ADCR0.

ADSR0, 1 (AD Setting Register) cu 3 biți comandă timpul de eșantionare (4,6,8,12,24,36,48 sau 128 cicluri mașină) și care dă un timp de eșantionare funcție de tact. Timpul de eșantionare trebuie ales în funcție de rezistența exterioară R_{ext} . Cu 3 biți se comandă timpul de comparare (22,33,44,66,88,132,176,264 cicluri mașină) care trebuie ales în funcție de V_{cc} . 5 biți (ANS4-ANS0) stabilesc canalul de start pentru conversia AD multiplă iar 5 biți (ANE4-ANE0) stabilesc ultimul canal pentru conversia AD multiplă. Dacă ANS4-ANS0 și ANE4-ANE0 indică același canal atunci conversia va fi simplă (individuală, continuă sau stop).

Observație

Modulul de conversie poate fi alimentat separat sau poate fi alimentat de la aceeași sursă, cu măsuri suplimentare de filtrare și stabilizare. Pinul de alimentare este V_{cc} , masa este V_{ss} iar tensiunea de referință este V_{RM} .

Efectuarea conversiei AD

1.Mod individual, intrările analogice ale canalelor începând cu cel definit de biții ANS și terminând cu cel definit de ANE sunt convertite, apoi AD se oprește (conversie multiplă individuală). Dacă s-a definit $ANS=ANE$ se convertește un singur canal (conversie simplă individuală).

2.Mod continuu, conversia are loc ca la modul individual dar după o conversie a canalelor selectate secvența se reia (conversie multiplă continuă). Pentru a forța oprirea convertorului se poate scrie bitul BUSY din ADCS1.

3.Mod Stop, se convertește un canal apoi AD intră în stop și așteaptă o activare (un semnal de Start). Conversia continuă pentru următorul canal, începând cu cel definit de ANS și terminând cu cel definit de ANE. Apoi conversia se reia.

5.2.2.Particularități ale convertorului AD la diferite familii de MC

Particularități la familia MB95xxxx

Convertorul care este integrat în MC din această familie este mai simplu și nu are atât de multe moduri de funcționare. Câteva caracteristici prin care diferă de convertorul AD prezentat anterior:

- Timpul de conversie este mai mare de $4,6\mu s$ (la un tact de 10MHz)
- Nu este posibil modul de conversie multiplă
- Nu este posibilă conversia continuă. O conversie repetată se poate realiza dacă bitul de declanșare software a conversiei se rescrie (chiar în timpul conversiei) sau dacă se activează modul de declanșare a conversiei de la un pin extern și acesta pornește repetat conversia pe front crescător. Conversia mai poate fi declanșată și de la un timer intern.
- Ca moduri de transfer a datelor nu se pot folosi transferuri pe blocuri de date (DMA)

Regiștrii

ADC1 (AD Control Register) cu 4 biți selectează unul dintre cele 16 canale analogice de intrare, un bit arată terminarea unei conversii și poate fi folosit la cererea unei întreruperi, un bit arată că AD are o conversie în curs, un bit declanșează soft o conversie.

ADC2 (AD Control Register) selectează rezoluția pe 8 sau 10 biți, 2 biți selectează timpul de eșantionare, un bit selectează pornirea externă, un bit validează tactul convertorului.

Pe lângă acești 2 regiștrii de control mai există doar regiștrii de date **ADDL** (AD Data Register Low) care stochează 8 biți mai puțin semnificativi ai rezultatului conversiei și **ADDH** (AD Data Register High) care stochează 2 biți mai semnificativi.

Particularități la MB91360

Funcționarea, structura și modul de programare sunt la fel cu cele descrise la familia pe 16 biți. Timpul de conversie este mai mare de 5,6μs.

Conversia continuă se folosește în combinație cu transferul DMA. O schemă logică simplificată a acestei operații este, figura 5.15:

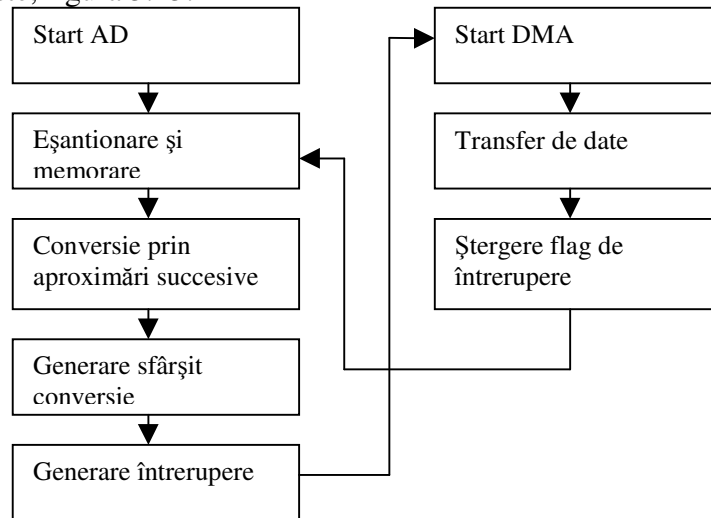


Figura 5.15: schema logică a conversiei continue

Pentru a evita ca datele să fie scrise în registrul de date înainte ca cele precedente să fie citite se introduce un mecanism de protecție, valabil la lucrul în întreruperi cu DMA. Flagul de întrerupere nu poate fi șters înainte de a utiliza un transfer DMA. Convertorul AD este introdus într-o buclă inactivă (PAUSE) până la citirea datelor.

Particularități la MB90350. Transferul datelor prin EI²OS

Schema logică a unei conversii continue combinate cu transferul EI²OS este dată în figura 5.16.

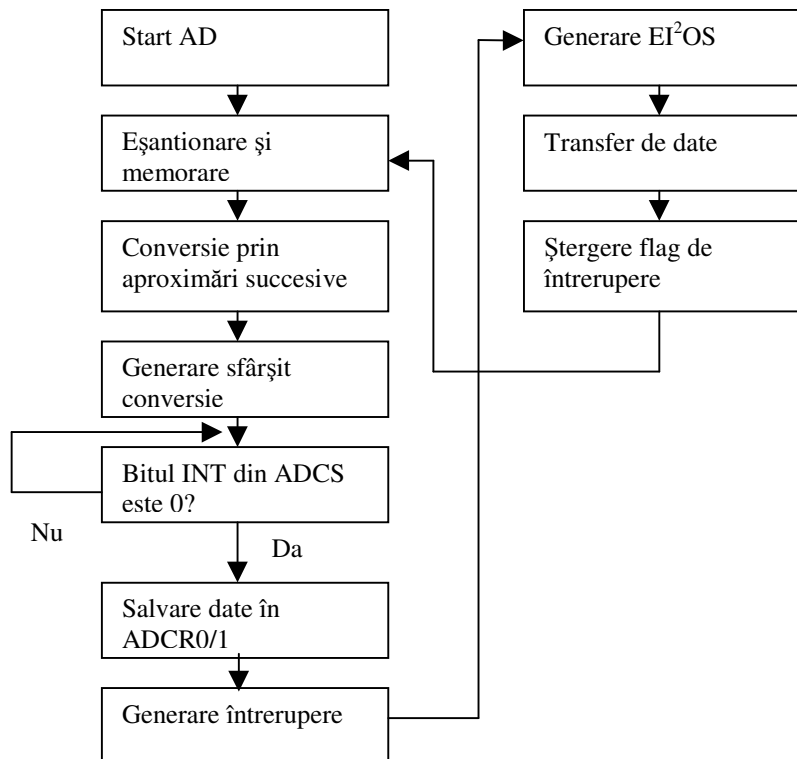


Figura 5.16: schema logică a conversiei cu transfer EI²OS

Se remarcă structura asemănătoare cu cea a transferului DMA.

Un exemplu de activare a modului EI²OS în mod conversie individuală multiplă este:

- Se convertesc datele AN1-AN3 și apoi conversia se termină
- Datele se stochează la adresele 200H-205H
- Activarea conversiei se face soft
- Se atribuie EI²OS prioritatea maximă.

Programul pentru acest exemplu este:

Setare EI ² OS	MOV ICR03, #08H	ICR03 Interrupt Control Register pentru timerul cu reîncărcare care activează EI ² OS: validare EI ² OS cu bitul ISE, cea mai mare prioritate (IL2,IL1,IL0 =0), selectare canal 0 EI ² OS
	MOV BAPL, #00H	BAPL Buffer Address Pointer Low, adresa de destinație a datelor (Low)
	MOV BAPM, #02H	BAPL Buffer Address Pointer Middle, adresa de destinație a datelor (Middle)
	MOV BAPH, #00H	BAPH BAPL Buffer Address Pointer High, adresa de destinație a datelor (High)
	MOV ISCS, #18H	EI ² OS Status Register: se comandă transfer pe 2 octeți, apoi incrementarea adresei. Direcția transferului este de la IO spre memorie
	MOV IOAL, #6AH	IO Address (Low) adresa portului de date AD, situată în zona de memorie IO (este găsită în tabele, de obicei în anexe)
	MOV IOAH, #00H	IO Address (High) adresa portului de date AD
	MOV DCTL, #03H	Data counter (Low) numărul de repetări ale operației (achiziției), 3 în acest caz
	MOV DCTH, #00H	Data counter (High) numărul de repetări ale operației
Setare convertor AD	MOV ADCS0, #00H	AD Control Status Register: mod conversie individuală
	MOVW ADSR0, #6823	AD Setting Register timpul de eșantionare se alege 600ns, timpul de comparare 2200ns (la un tact de 20MHz), canalul de start AN1, ultimul canal AN3
	MOV ADCS1, #A2H	AD Control Status Register, start conversie
Secvența de întrerupere	MOV ADCS1, #80H	AD Control Status Register, în ADC se anulează cererea de întrerupere pentru a fi posibilă o nouă conversie
	RETI	

Transferul are loc conform cu următoarea diagramă simplificată care detaliază mecanismul EI²OS, figura 5.17:

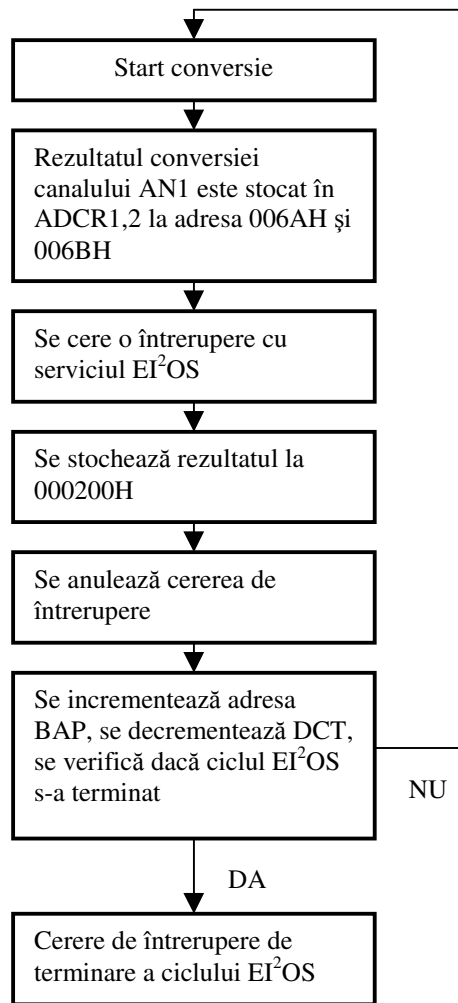


Figura 5.17: mecanismul EI²OS

5.2.3. Convertorul DA

Este descris convertorul integrat în MC din familia MB95xxxx, cu rezoluția de 10 biți, construit cu rețea R-2R. Plaja valorilor de ieșire este de la 0V la 1024xAV_{cc}, așa încât se poate plaja se poate modifica schimbând tensiunea de alimentare a părții analogice a MC.

Schema bloc a convertorului DA este dată în figura 5.18.

Regiștrii

DAT (DA Converter Data Register) conține cei 8 biți mai puțin semnificativi ai cuvântului de convertit.

DACR (DA Converter Control Register) conține cei 2 biți mai puțin semnificativi ai cuvântului de convertit, validarea ieșirii (DAE) și selecția pinului de ieșire ca și pin de I/O de uz general sau ieșire analogică.

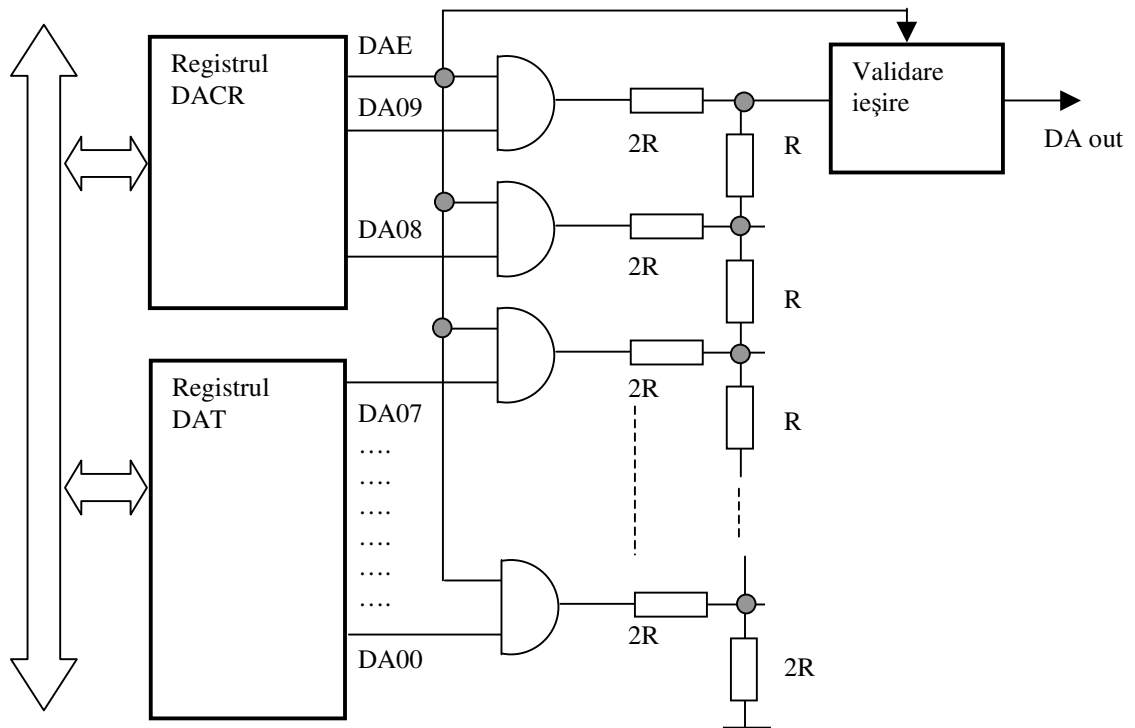


Figura 5.18: schema bloc a convertorului D/A

Se recomandă următoarea procedură pentru conversia datelor:

- Se încarcă registrul DACR cu cei 2 biți mai semnificativi, semnalul DAE fiind 0 (ieșire invalidată)
- Se încarcă DAT cu cei 8 biți mai puțin semnificativi
- Se validează ieșirea prin încărcarea DACR cu DAE=1, cei 2 biți mai semnificativi fiind păstrați
- Se schimbă datele de convertit dacă este nevoie, a se vedea observația 2.

Observația 1

Dacă se validează ieșirea cu DAE=1 din primul pas, ieșirea analogică devine 0, ceea ce înseamnă o valoare eronată.

Observația 2

Pentru a actualiza cei 2 biți mai semnificativi trebuie să se reîncarce și cei 8 biți mai puțin semnificativi. Datele de convertit se schimbă astfel:

- Se încarcă DACR cu cei 2 biți mai semnificativi (DAE=1)
- Se încarcă DAT cu cei 8 biți mai puțin semnificativi (chiar dacă aceștia au rămas neschimbați de la conversia anterioară).

Observația 3

La intrarea în modurile cu economie de energie, ieșirea păstrează valoarea analogică comandată. Pentru micșorarea consumului, dacă nu este nevoie de păstrarea acestei valori, DAE trebuie înscris cu 0.

5.3. Interfața I²C

Interfața I²C poate fi integrată în MC doar cu plata drepturilor de autor către Philips. De aceea nu toate MC au integrată această interfață, doar cele cu sufixul C.

5.3.1. Principiul interfeței I²C

Interfața (magistrala) serială I²C (Inter IC Bus) a fost dezvoltată de Philips. Distanța de comunicare prin această interfață este scurtă iar imunitatea la perturbații este medie. Această interfață a fost dezvoltată pentru a interconecta aparate de uz casnic, circuite integrate etc., comandate de microprocesor sau microcontroller.

Interfața constă în 2 linii: SDA (Date Seriale) și SCL (Tact Serial). Ea permite viteze de până la 100Kbps respectiv 400Kbps (sunt 2 variante de I²C) și are un mecanism de arbitrare master/ slave. Cele 2 linii SDA și SCL sunt realizate prin conectarea ieșirilor /intrărilor tuturor aparatelor conectate prin porți cu colectorul în gol. Fiecare linie este ținută la 1 logic printr-o rezistență legată la +5V. Transmisia pe linii se face cu nivele TTL iar codificarea datelor este NRZ.

Bitul de date este citit / scris cu un bit de tact (figura 5.15), iar o anumită combinație a biților de date și tact determină startul și o alta oprirea transferului, figura 5.19:

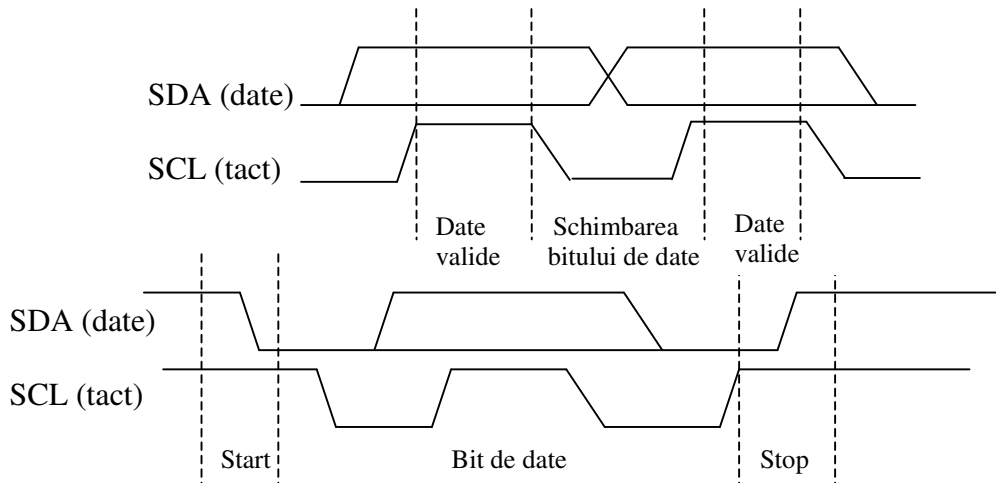


Figura 5.19: bit de date și tact, condiția de Start și Stop

După o combinație de Start magistrala este considerată ocupată (BUSY). După combinația de Stop magistrala este considerată liberă. Biții de date/ adrese se trimit seria pe SDA începând cu cel mai semnificativ bit.

Sincronizarea transferului

a. La nivel de octet: dispozitivul slave trebuie să răspundă după primirea unui octet prin stabilirea unui bit ACK. Dacă acest bit este 0 transferul poate continua, dacă este 1 transferul se abandonează. Bitul ACK este al 9-lea bit transmis pe linie, figura 5.20:

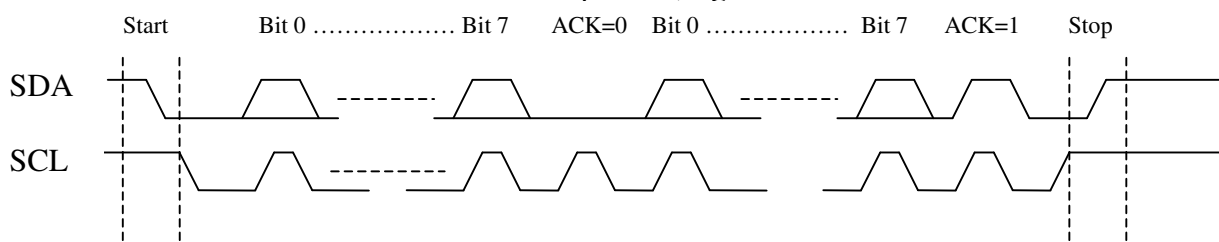


Figura 5.20: sincronizarea transferului

Folosirea bitului ACK este obligatorie.

b.La nivel de bit: un alt nivel de sincronizare, între un dispozitiv master și un slave mai lent decurge astfel: pentru transferul unui bit dispozitivul master pune SCL în 0, apoi în 1. Dispozitivul slave lent poate ține linia SCL în 0 până poate accepta sau transmite date.

Arbitrarea magistralei

Se face printr-un protocol de detecție a coliziunilor ca și la rețeaua Ethernet. Un dispozitiv master care vrea să preia controlul magistralei verifică dacă este liberă, adică dacă SDA și SCL sunt în stare H. Timpul minim în care trebuie să fie acestea în stare H este dat de specificațiile de magistrală și este de $4,7\mu\text{s}$ (ambele în stare H sunt și în cadrul unui transfer de date, dar o perioadă de timp mai mică). Dacă magistrala este liberă dispozitivul master generează o secvență de start după care pune pe linie cei 7 (10) biți care semnifică adresa echipamentului definit ca slave. Dacă 2 dispozitive master solicită simultan magistrala are loc o coliziune care este detectată, figura 5.21:

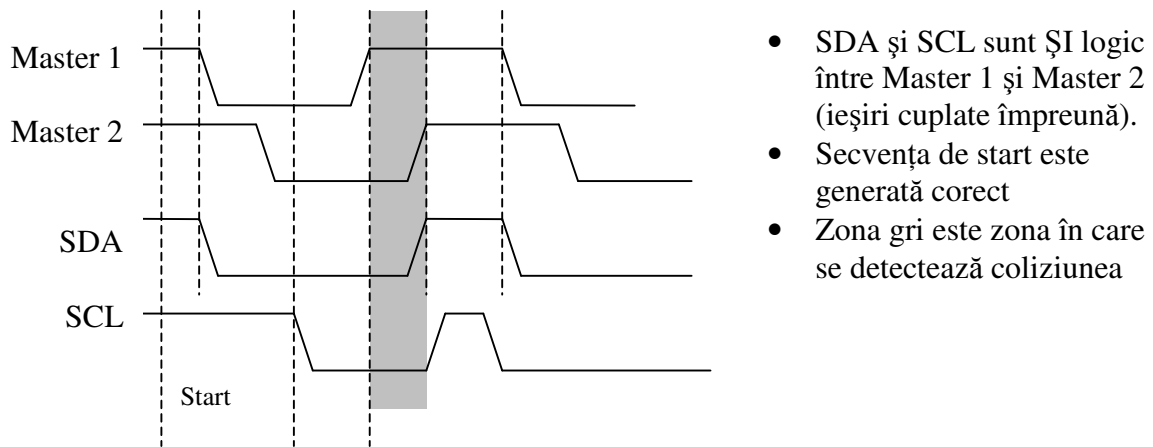


Figura 5.21: detectarea coliziunii

Fiecare dispozitiv master, ca să transmită un 1 logic lasă liberă linia SDA. În același timp el citește linia SDA. Dacă ea este în 0 înseamnă că un alt master transmite și astfel s-a detectat o coliziune. Coliziunea este detectată de masterul care transmite un 1 prima dată. După detectarea coliziunii transferul continuă între dispozitivul care nu a detectat coliziunea și dispozitivul slave, iar masterul care a detectat coliziunea abandonează transferul.

Adresarea

Adresarea se face pe 7 biți, deci maximum 128 de echipamente cuplate. Dintre aceste adrese nu toate sunt libere. Philips a extins numărul de biți de adresă la 10, cu trimiterea adresei în 2 octeți. Adresa dispozitivului slave este pusă de master și constă în 7 biți din care primul este cel mai semnificativ. Al optulea bit arată sensul transferului solicitat (Read sau Write). Adresele 00-0FH și F0-FFH sunt rezervate. Transmiterea adresei este urmată de recepționarea unui ACK pus de slave.

5.3.2. Interfața I²C de 400KHz

Această interfață echipează MC pe 16 și 32 de biți (MB90350 și MB91360). O interfață mai simplă care admite viteză mai mică (100KHz) și permite doar adresarea pe 7 biți echipează MC din familia MB95xxxx, dar și MC mai performante (MB91360) care au și canale de viteză mare și canale de viteză mică. Caracteristicile principale ale interfeței de 400KHz sunt:

Mod de transmisie serial master/ slave cu posibilități de arbitrare

- Adresare pe 7 sau 10 biți
- Rata de transfer de până la 400Kbps

- Filtre integrate de zgomot pe liniile de date și tact
- Suportă ca rata de transfer să fie micșorată de slave la nivel de bit și de octet.

Schema bloc simplificată a acestei interfețe este dată în figura 5.22:

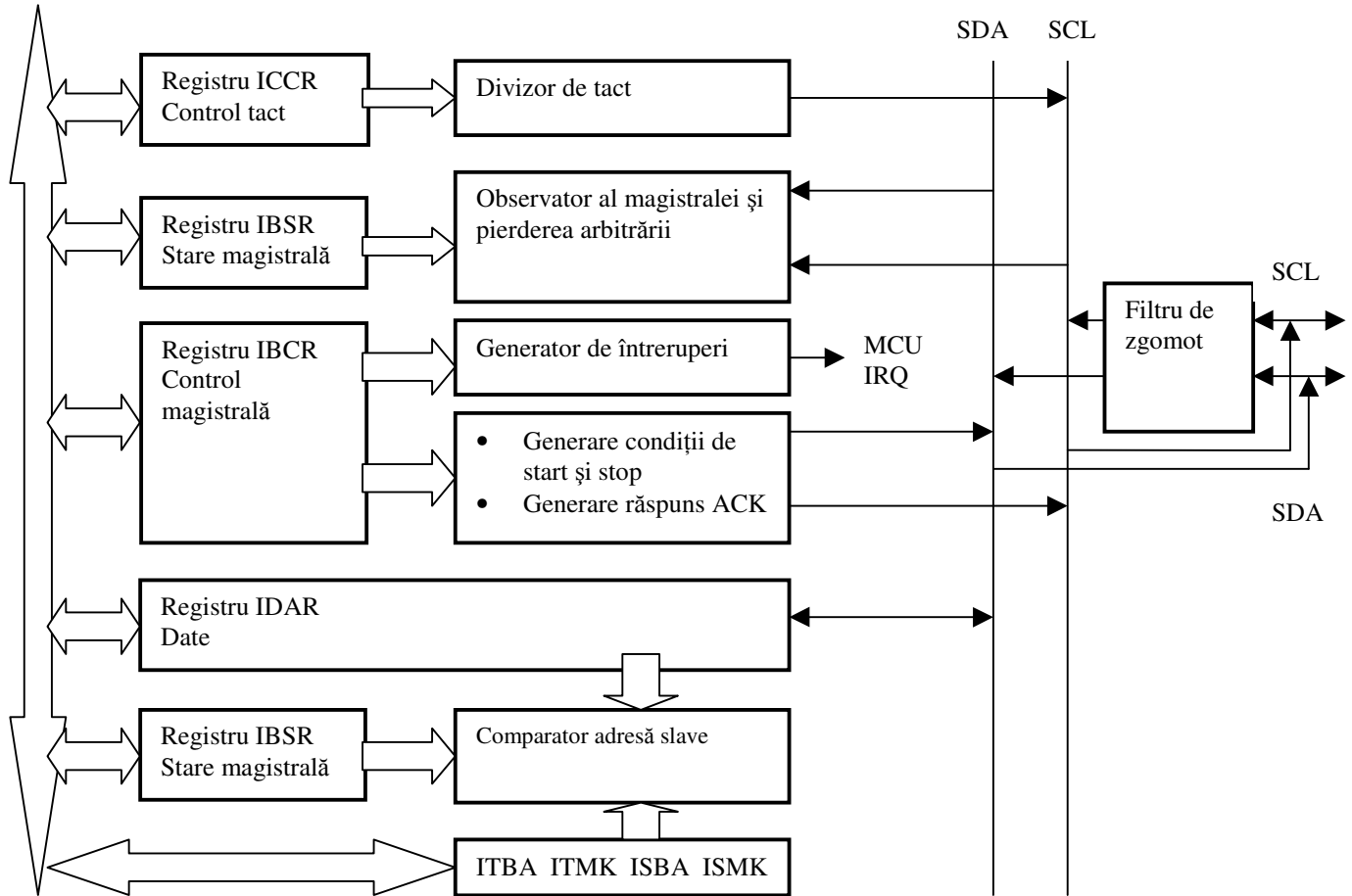


Figura 5.22: schema bloc a interfeței I²C

Transferurile se execută pe 2 linii SDA și SCL bidirecționale, cu colector în gol care permit conectarea directă a dispozitivelor pe magistrală. La intrare datele și tactul sunt filtrate pentru a elimina zgomotul (impulsurile scurte parazite). Datele recepționate în registrul de date IDAR sunt comparate cu adresa proprie a dispozitivului (pe 7 sau 10 biți). Adresa proprie este încărcată în comparator cu regiștrii ITBA (adresa pe 10 biți) și ISBA (adresa pe 7 biți). La detectarea adresei proprii se generează un ACK și începe recepționarea datelor. Datele de transmis se încarcă în registrul de date IDAR. Se generează o condiție de start și se trimite adresa slave-ului. La primirea ACK începe transferul datelor. După terminarea transferului se generează condiția de stop. Generatorul de întreruperi cere o întrerupere la recepționarea unui caracter, la transmisia unuia sau la apariția unei erori. Un modul supraveghează permanent magistrala pentru a detecta o coliziune, caz în care abandonează transferul. Un divizor de tact asigură tactul necesar transmisiei datelor.

Regiștrii

IBSR (Bus Status Register), un bit arată starea magistralei (în stop sau în operare), un bit arată dacă a fost detectat un restart, un bit arată pierderea arbitrării, un bit arată că receptorul a răspuns sau nu (ACK), un bit arată dacă se transmit date sau nu, un bit arată dacă interfața a fost adresată ca slave, un bit arată că s-a detectat o adresă de salt general (adresa 00H), un bit arată că se transferă o adresă.

Observație: adresa de salt general este o adresă care se transmite înainte de adresa slave-ului. O pierdere a arbitrării face ca totuși adresa de slave care urmează să fie corect transmisă și transferul poate avea loc.

IBCR (Bus Control Register), un flag de eroare care poate cere întrerupere, un bit validează această cerere de întrerupere, un bit comandă repetarea continuă a condiției de start, un bit stabilește dacă circuitul este master sau slave (MSS), un bit validează trimiterea răspunsului ACK la recepția unui octet, un bit validează trimiterea răspunsului ACK la recepția unei adrese de salt general, un bit validează sistemul de întreruperi, un bit cere întrerupere la terminarea unui transfer.

Observație: trecerea în mod master comandată de bitul MSS este urmată de generarea unei condiții de start, trimiterea adresei slave-ului și trimiterea sau recepționarea datelor.

ITBAH, ITBAL (Ten Bit Slave Address Register) conține adresa slave-ului pe 10 biți. Dacă o adresă este recepționată ea se compară cu ITBA (dacă este validat modul de lucru cu 10 biți de adresă în ITMK). Dacă adresele coincid se transmite un ACK și se permite începutul transferului de date.

ITMKH, ITMKL (Ten Bit Address Mask Register) conține o mască pe 10 biți pentru adresa slave-ului, un bit validează modul de lucru cu 10 biți de adresă iar un bit arată dacă adresa recepționată conține 7 sau 10 biți. În mască un bit de 0 face ca bitul respectiv din adresa recepționată să nu fie comparat cu bitul din ITBA. Se pot astfel atribui unui slave adrese multiple.

Observația 1: circuitul recunoaște dacă adresa este pe 10 biți prin faptul că primul octet de adresă este de forma FxH, combinație rezervată în adresarea pe 7 biți.

Observația 2: acești regiștrii lipsesc la varianta de interfață la 100KHz și adresare pe 7 biți.

ISBA (Seven Bit Slave Address Register) conține adresa slave-ului pe 7 biți.

ISMK (Seven Bit Address Mask Register) conține o mască pe 7 biți pentru adresa slave-ului, un bit validează modul de lucru cu 7 biți de adresă. În mască un bit de 0 face ca bitul respectiv din adresa recepționată să nu fie comparat cu bitul din ISBA, însă nu se poate lucra cu adrese multiple.

IDAR (Data Register) conține datele pe 8 biți, începând cu MSB. Registrul este dublu buffer-at la scriere pentru ca datele de trimis să poată fi stocate în acest registru în timp ce octetul anterior se transmite. La citire registrul trebuie citit doar după ce bitul de cerere de întrerupere din IBCR arată terminarea transferului.

ICCR (Clock Control Register), un bit validează filtrele pe liniile de date și tact care elimină impulsurile mai scurte de 1-1,5 cicluri mașină. Filtrele trebuie invalidate la 100Kbps. Un bit validează interfața, 5 biți selectează rata de transfer printr-un coeficient de prescalare n. Rata de transfer este funcție de tact, astfel la un tact de 24MHz și n=4 se obține o rată de transfer de 369Kbps.

Operarea I²C

Condiția de start se generează astfel:

- Dacă magistrala este liberă (un bit din IBSR arată acest lucru) se poziționează bitul de master în IBCR (bitul MSS) și se trimite adresa slave-ului stocată în IDAR pe linia I²C.
- Dacă se trimit date poziționând bitul de master din IBCR și magistrala nu este liberă, se așteaptă eliberarea ei.
- Dacă interfața este slave și recepționează date, dar are și de transmis, atunci așteaptă terminarea transferului de recepție și eliberarea magistralei.

Condiția de stop se generează prin resetarea bitului MSS în IBCR.

Detectarea adresei de slave se realizează astfel:

În mod slave datele sunt recepționate în IDAR.

- Datele din IDAR sunt comparate cu cele din ISBA ținând cont de masca din ISMK și dacă sunt egale se răspunde cu ACK.
- Dacă este validată adresarea pe 10 biți datele din IDAR sunt comparate cu cele din ITBA ținând cont de masca din ITMK.

În mod master adresarea dispozitivelor slave se face punând în IDAR adresa slave-ului și generând apoi o condiție de start cu bitul MSS din IBCR, apoi se așteaptă răspunsul ACK al dispozitivului slave. Adresele se trimit astfel:

- Adresă de 7 biți, acces la scriere: A6 A5 A4 A3 A2 A1 A0 0 (un octet)
- Adresă de 7 biți, acces la citire: A6 A5 A4 A3 A2 A1 A0 1 (un octet)
- Adresă de 10 biți, acces la scriere: 1 1 1 1 0 A9 A8 0 A7 A6 A5 A4 A3 A2 A1 A0 0 (2 octeți)
- Adresă de 10 biți, acces la citire: 1 1 1 1 0 A9 A8 1 A7 A6 A5 A4 A3 A2 A1 A0 0 (2 octeți)

Observație: un bit din adresă arată sensul transferului solicitat.

În timpul transmisiei în mod master, dacă un alt master dorește transmisia se realizează o arbitrare. Arbitrarea este declanșată de de trimiterea unei valori de 1 pe linia SDA care are valoarea logică 0. Modulul de observare a magistralei constată nivelul 0 al liniei SDA și consideră că s-a pierdut arbitrarea, ca urmare poziționează bitul corespunzător în IBSR. În acest caz interfața trece în mod slave și citește adresele de pe linie pentru a verifica dacă este adresată.

Confirmarea ACK este trimisă de la receptor către transmițător, dacă este validată această confirmare în IBCR. Recepția unei confirmări poate fi verificată printr-un bit din IBSR.

Pentru exemplificare se arată în diagrama de timp următoare (figura 5.23) un transfer I²C de la master la slave cu adresă de 7 biți, în ipoteza că magistrala este liberă:

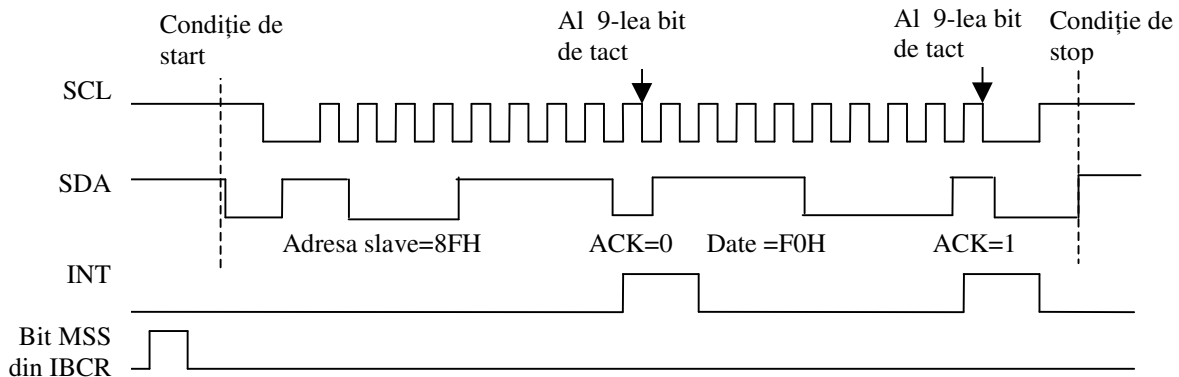
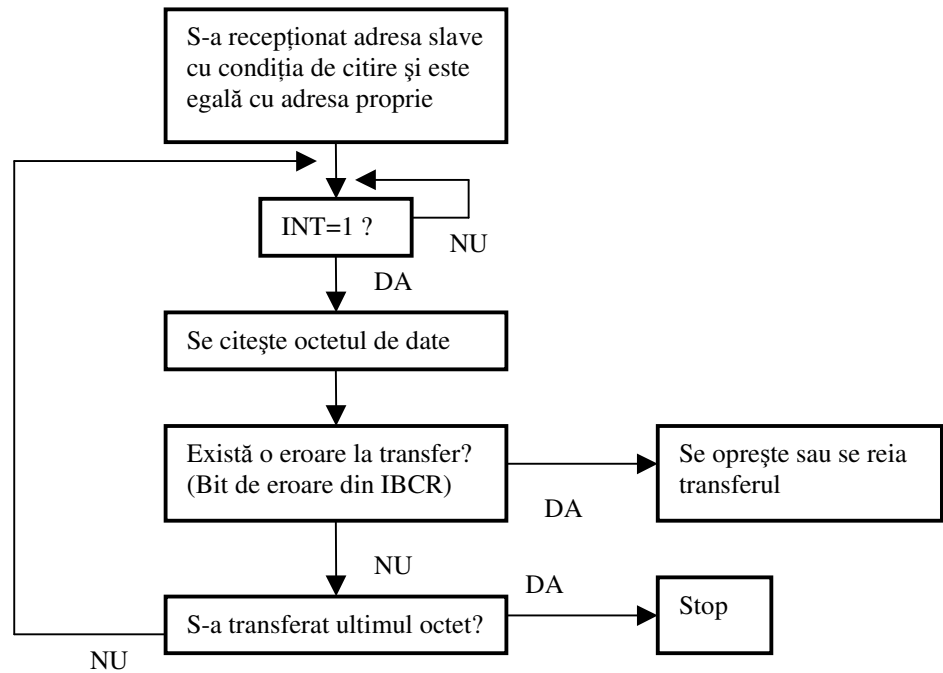


Figura 5.23: diagrama unui transfer I²C de la master la slave

Funcționarea ca slave pentru recepția datelor poate fi urmărită în următoarea diagramă simplificată, figura 5.24. În această diagramă nu s-a reprezentat recunoașterea adresei proprii și generarea ACK. O diagramă simplificată a funcționării ca slave pentru trimiterea datelor este dată în aceeași figură.

Funcționarea ca slave pentru recepția datelor



Funcționarea ca slave pentru trimiterea datelor

Această diagramă este mai detaliată și conține verificările care se fac la trimiterea datelor de la slave către master:

- Dacă INT devine activ înseamnă că bufferul de emisie a devenit gol prin trimiterea octetului pe linie și se poate trimite următorul octet
- Dacă bitul de eroare din IBCR semnalizează o eroare se reia transferul
- Dacă se pierde arbitrarea (se detectează o coliziune) se abandonează transferul
- Dacă master-ul nu răspunde cu ACK că a recepționat datele se termină transferul.

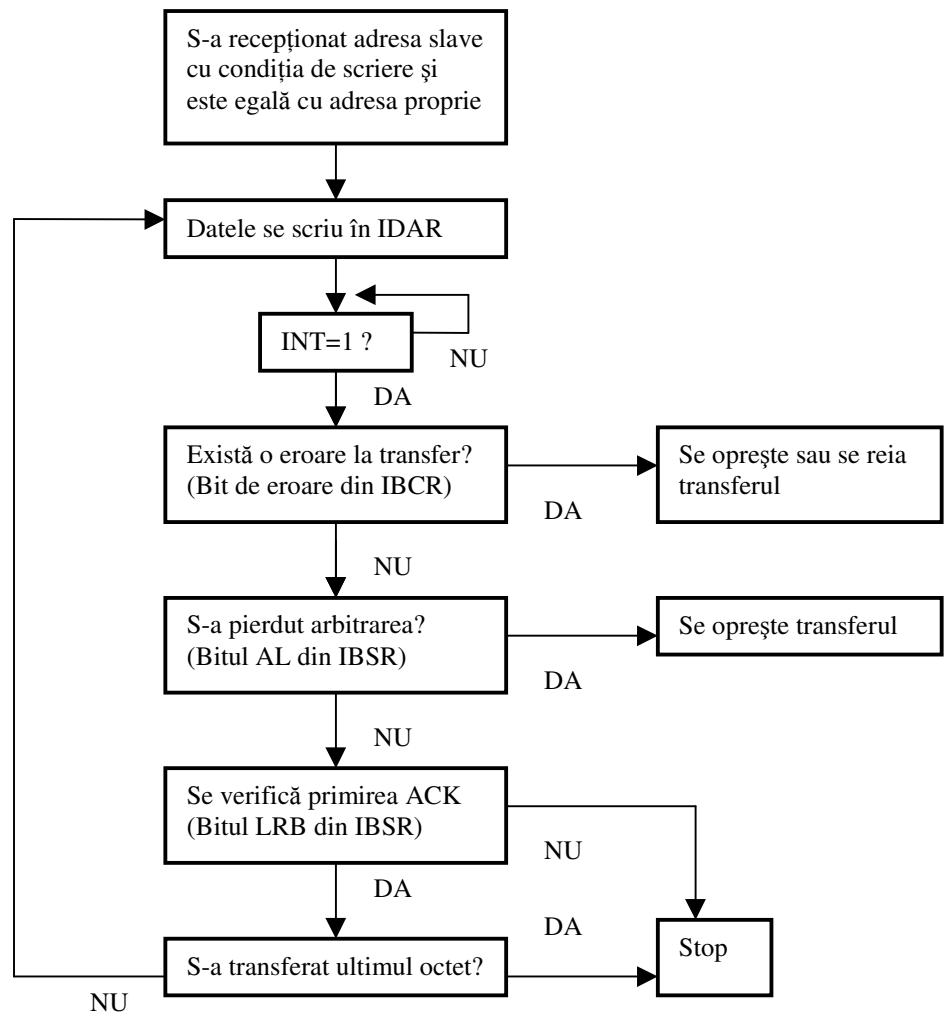


Figura 5.24: scheme logice