

PERIFERICE ALE UNUI MICROCONTROLLER DIN FAMILIA HCS12X

4.1. INTRODUCERE

În acest capitol se prezintă structura și funcționarea unor periferice ale microcontrollerului MC9S12XDP512 din familia HCS12X:

- porturile paralele;
- modulul temporizator pentru întreruperi periodice;
- modulul temporizator cu funcții de numărare, captură și comparare;
- modulul generator de impulsuri modulate în durată;
- modulul de conversie analog numerică;
- modulul de interfață pentru comunicație serială asincronă.

4.2. PORTURILE PARALELE

Un microcontroller din familia HCS12X conține 14 porturi paralele de câte 8 biți, fig. 4.1. Fiecare port poate fi utilizat pentru transfer de date de uz general sau pentru interfața între perifericele interne ale microcontrollerului și componente externe:

- porturile A și B pentru magistrala de adrese ADDR15÷ADDR0 a interfeței externe EBI (*External Bus Interface*);
- porturile C și D pentru magistrala de date DATA15÷DATA0 a interfeței externe;
- portul E pentru semnale de control ale interfeței externe și cereri de întrerupere /IRQ, /XIRQ;
- portul K pentru liniile de adrese ADDR22÷ADDR16 ale interfeței externe și un semnal de control al interfeței externe;
- portul T pentru interfața cu modulul temporizator cu funcții de captură și comparare ECT (*Enhanced Capture Timer*);
- portul S pentru două interfețe de comunicație serială asincronă SCI (*Serial Communication Interface*) și o interfață de comunicație serială sincronă SPI (*Serial Peripheral Interface*);
- portul M pentru patru porturi seriale de comunicație conform protocolului CAN (*Controller Area Network*) și o interfață de comunicație serială asincronă SCI;

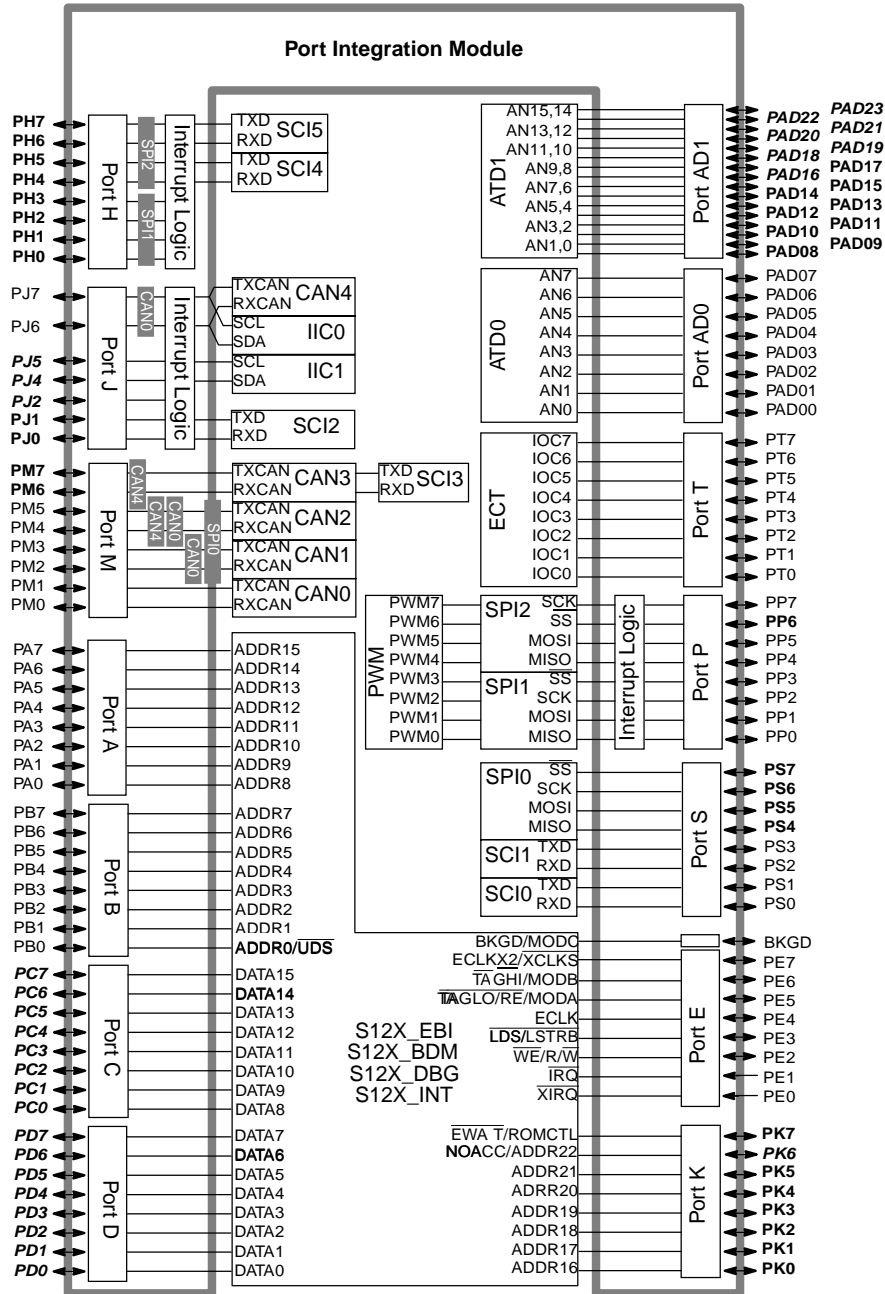


Fig. 4.1. Porturile paralele ale unui microcontroler din familia HCS12X.

- portul P pentru modulatorul de impulsuri în durată PWM (*Pulse-Width Modulator*) și două interfețe de comunicație serială sincronă SPI;
- portul H pentru două interfețe de comunicație serială asincronă SCI;
- portul J pentru un port serial de comunicație conform protocolului CAN, o interfață de comunicație serială asincronă SCI și două porturi seriale IIC (*Inter-Integrated Circuit*);
- porturile AD0 și AD1 pentru cele două convertoare analog numerice ATD (*Analog-to-Digital Converter*).

Funcționarea și programarea porturilor paralele se bazează pe utilizarea unor registre conform tabelului 4.1.

Fiecare port conține un **registru de date** **PORT_** (*Port _ Data Register*) și un **registru de direcție date** **DDR_** (*Port _ Data Direction Register*) utilizabile pentru transfer de date de uz general. Prin biții registrului de direcție date se programează individual pinii corespunzători portului ca ieșiri sau intrări. Nivelul logic 1 al unui bit din registrul de direcție programează pinul corespunzător ca ieșire, iar nivelul logic 0 al unui bit din registrul de direcție programează pinul corespunzător ca intrare. Pinii de ieșire sunt comandați de biții corespunzători ai registrului de date. Nivelurile logice de comandă externă a pinilor de intrare se transferă în biții corespunzători ai registrului de date, fig. 4.2.

Registrul de intrare **PTI_** (*Port _ Input Register*) al unui port este accesibil numai pentru citire și conține nivelurile logice de la pinii portului, fig. 4.2.

Tabelul 4.1

Port	Data	Data Direction	Input	Reduced Drive	Pull Enable	Polarity Select	Wired-OR Mode	Interrupt Enable	Interrupt Flag
A	yes	yes	—	yes	yes	—	—	—	—
B	yes	yes	—			—	—	—	—
C	yes	yes	—			—	—	—	—
D	yes	yes	—			—	—	—	—
E	yes	yes	—			—	—	—	—
K	yes	yes	—			—	—	—	—
T	yes	yes	yes	yes	yes	—	—	—	—
S	yes	yes	yes	yes	yes	yes	yes	—	—
M	yes	yes	yes	yes	yes	yes	yes	—	—
P	yes	yes	yes	yes	yes	yes	—	yes	yes
H	yes	yes	yes	yes	yes	yes	—	yes	yes
J	yes	yes	yes	yes	yes	yes	—	yes	yes
AD0	yes	yes	—	yes	yes	—	—	—	—
AD1	yes	yes	—	yes	yes	—	—	—	—

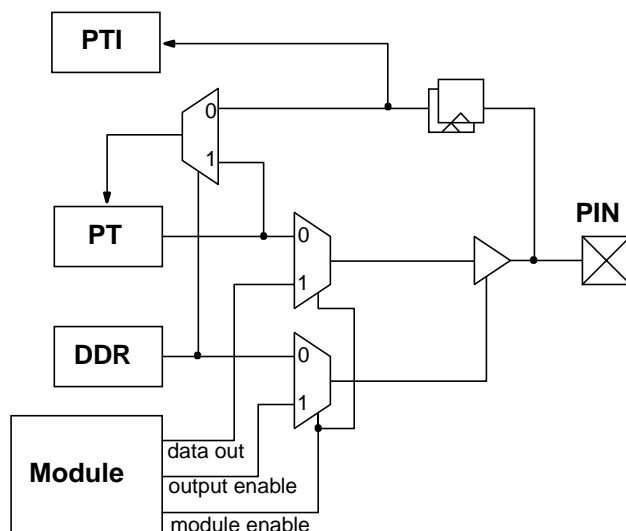


Fig. 4.2. Structura funcțională a unui port paralel corespunzătoare unei linii de intrare/ieșire.

În figura 4.2 sunt utilizate notațiile:

Module - periferic al microcontrolerului care poate comanda pinul PIN;

PIN - pin al microcontrolerului corespunzător portului paralel;

DDR - bit al registrului de direcție date;

PT - bit al registrului de date;

PTI - bit al registrului de intrare.

Registrul comandă redusă RDR_o (*Port _ Reduced Drive Register*) al unui port permite programarea puterii de comandă pentru pinii de ieșire. Nivelul logic 0 al unui bit din acest registru determină funcționarea la putere de comandă totală disponibilă, iar nivelul logic 1 determină reducerea de 6 ori a puterii de comandă la pinul corespunzător. Pentru porturile A, B, C, D, E și K utilizate la interfața externă există un registru comun RDRIV prin biții căruia se poate programa putere de comandă redusă la nivel de port și nu la nivel de bit ca în cazul celorlalte porturi.

Registreele PER_o (*Port _ Pull Device Enable Register*) și **PPS_o** (*Port _ Polarity Select Register*) ale unui port validează și selectează conectarea internă la pinii portului a unor dispozitive de tip *pull-up* sau *pull-down*. Această validare este activă numai pentru pinii de intrare.

Registrul WOM_o (*Port _ Wired-OR Mode Register*) al unui port permite deconectarea sarcinilor active (*active high drive*) pentru pinii de ieșire în scopul realizării unor conexiuni de ieșiri în varianta SAU-cablat.

Registrul de validare întreruperi PIE_ (*Port _ Interrupt Enable Register*) validează/invalidează cererile de întrerupere cu fronturi de ridicare sau coborâre de la pinii portului. În cazul validării acestor întreruperi, fronturile active se selectează la nivel de pin cu registrul PPS_. Fiecărui port prin care se pot face cereri de întrerupere îi corespunde un singur vector de întrerupere.

Registrul indicator de întreruperi PIF_ (*Port _ Interrupt Flag Register*) memorează cererile de întrerupere de la pinii portului pentru care sunt validate cererile de întrerupere cu registrul PIE.

4.3. MODULUL TEMPORIZATOR PENTRU ÎNTRERUPERI PERIODICE

Funcționarea de principiu a unui temporizator

Un temporizator generează evenimente periodice concretizate prin semnale și/sau cereri de întrerupere. Perioada evenimentelor (temporizarea) se stabilește prin numărarea unui anumit număr de impulsuri de frecvență dată. Astfel, temporizatorul conține un numărător al cărui conținut se decrementează cu câte o unitate la fiecare impuls de intrare. Temporizarea (evenimentul) corespunde anulării conținutului numărătorului. Primul impuls de intrare după anularea conținutului numărătorului declanșează încărcarea numărătorului cu o constantă de timp CT memorată într-un registru și a cărei valoare determină durata temporizării. După încărcarea constantei de timp se continuă cu decrementarea conținutului numărătorului rezultând temporizări periodice. Procedura prezentată corespunde funcționării unui numărător modulo $(CT+1)$ cu decrementare (*modulus down-counter*). Perioada temporizării este $(CT+1)/f$, unde f este frecvența semnalului de la intrarea numărătorului.

Aplicație

Se consideră un temporizator cu numărător de $N=16$ biți comandat la intrare cu un semnal de tact cu frecvența $f=8$ MHz. Astfel, constanta CT poate avea valori în intervalul $0 \div 2^{16}-1$, corespunzătoare unor temporizări în intervalul $(1 \cdot 125) \text{ ns} \div (2^{16} \cdot 125) \text{ ns} = 8,192 \text{ ms}$.

Modulul temporizator pentru întreruperi periodice **PIT** (*Periodic Interrupt Timer*), fig. 4.3, conține 4 canale cu temporizări programabile.

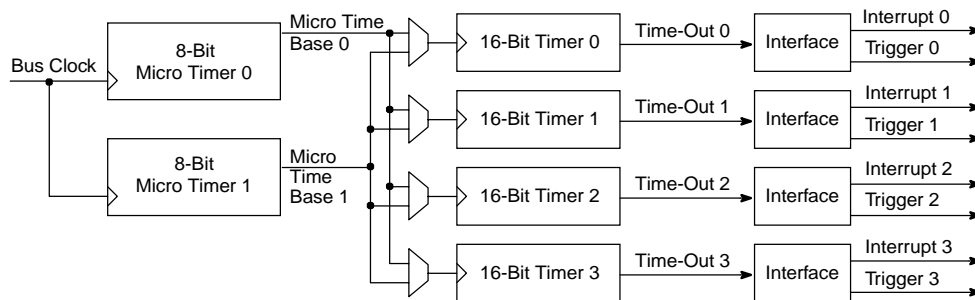


Fig. 4.3. Structura simplificată a modulului temporizator pentru întreruperi periodice.

Fiecare canal este realizat prin conectarea în serie a unui temporizator de 16 biți (16-bit *Timer* 0÷3) și a unui temporizator de 8 biți (8-bit *Micro Timer* 0 sau 1). Temporizatorul de 8 biți numără impulsuri cu frecvența ciclurilor f_{BUS} și comandă intrarea de numărare a temporizatorului de 16 biți. Temporizarea (evenimentul)

corespunde anulării conținuturilor celor două numărătoare. Prin circuitele de interfață (*interface*), fiecare canal poate să declanșeze un modul periferic al microcontrolerului sau să genereze cereri de întrerupere, cu perioadă programabilă (perioada temporizării) în intervalul $1 \div 2^{24}$ perioade ale semnalului f_{BUS} . Pentru $f_{BUS}=8$ MHz, perioada maximă a temporizării este de $256 \cdot 65536 \cdot 125 \text{ ns} = 2,097 \text{ s}$.

Structura detaliată a modului temporizator pentru întreruperi periodice, fig. 4.4, indică registrele utilizate pentru funcționarea și programarea modului PIT. În cele ce urmează se prezintă funcțiile acestor registre.

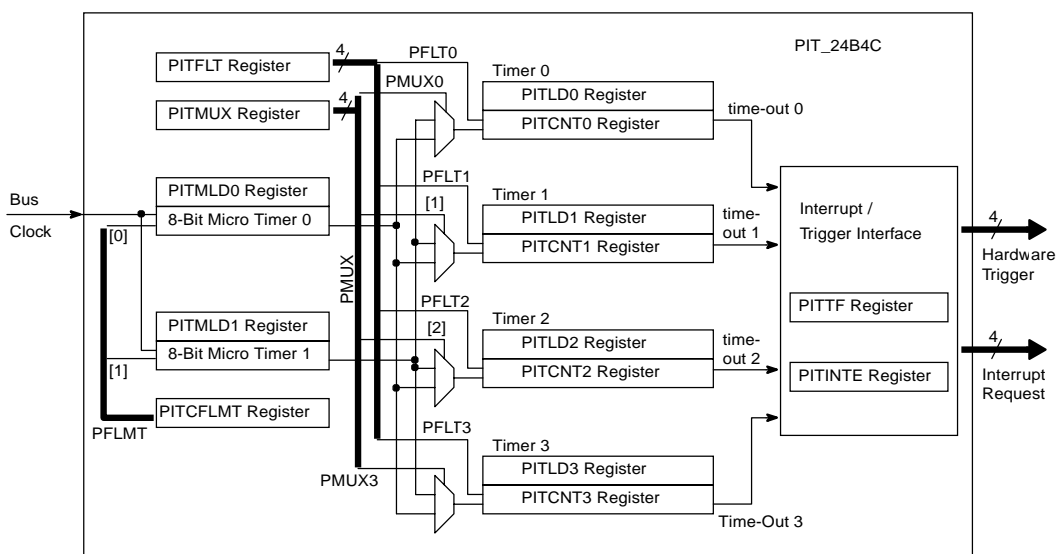


Fig. 4.4. Structura detaliată a modului temporizator pentru întreruperi periodice.

Registrul PITCFLMT (*PIT Control and Force Load Micro Timer Register*), 0340h.

Bitul **PITE** (PITCFLMT.7), la nivel logic 1/0, validează/invalidază modulul PIT.

Biții **PFLMT** $1 \div 0$ (PITCFLMT.1 $\div 0$) corespund temporizatoarelor de 8 biți (8-bit *Micro Timer*) 1 respectiv 0. Scrierea nivelului logic 1 în bitul **PFLMT** x determină încărcarea constantei de timp din registrul **PITMLD** x în numărătorul temporizatorului x de 8 biți.

Registrul PITFLT (*PIT Force Load Timer Register*), 0341h.

Biții **PFLT** $3 \div 0$ (PITFLT.3 $\div 0$) corespund temporizatoarelor de 16 biți (16-bit *Timer*) 3 $\div 0$. Scrierea nivelului logic 1 în bitul **PFLT** x determină încărcarea constantei de timp din registrul **PITLD** x în numărătorul **PITCNT** x al temporizatorului x de 16 biți.

Registrul PITCE (*PIT Channel Enable Register*), 0342h.

Biții **PCE** $3 \div 0$ (PITCE.3 $\div 0$) corespund temporizatoarelor de 16 biți (16-bit *Timer*) 3 $\div 0$. La nivel logic 1, bitul **PCE** x validează canalul x , dacă modulul PIT este validat cu bitul **PITE**. Validarea înseamnă încărcarea constantelor de timp și declanșarea procesului de numărare prin decrementare.

Registrul PITMUX (PIT Multiplex Register), 0343h.

Biții $PMUX_{3÷0}$ ($PITMUX_{3÷0}$) corespund temporizatoarelor de 16 biți (16-bit Timer) $3÷0$. Bitul $PMUX_x$, la nivel logic 0/1, comandă conectarea la canalul x al modului PIT a temporizatorului de 8 biți (8-bit Micro Timer) 0/1.

Registrul PITINTE (PIT Interrupt Enable Register), 0344h.

Biții $PINTE_{3÷0}$ ($PITINTE_{3÷0}$) corespund temporizatoarelor de 16 biți (16-bit Timer) $3÷0$. La nivel logic 1, bitul $PINTE_x$ validează cererile de întrerupere de la canalul x al modului PIT.

Registrul PITTF (PIT Time-Out Flag Register), 0345h.

Biții $PTF_{3÷0}$ ($PITTF_{3÷0}$) corespund temporizatoarelor de 16 biți (16-bit Timer) $3÷0$. Bitul PTF_x se poziționează la nivel logic 1 în momentul (temporizării) anulării conținutului numărătorul $PITCNT_x$ al temporizatorului de 16 biți x . Bitul PTF_x se anulează prin scriere nivel logic 1.

Registrele PITMTLD0÷1 (PIT Micro Timer Load Register) 0346h÷0347h;

Registrele $PITMTLD_{0÷1}$ corespund temporizatoarelor de 8 biți (8-bit Micro Timer) $0÷1$. Registrul $PITMTLD_x$ se încarcă cu constanta de timp pentru temporizatorul x de 8 biți.

Registrele PITLD0÷3 (PIT Load Register), 0348h+4*x, x=0÷3.

Registrele $PITLD_{0÷3}$ de câte 16 biți corespund temporizatoarelor de 16 biți (16-bit Timer) $0÷3$. Registrul $PITLD_x$ se încarcă cu constanta de timp pentru temporizatorul x de 16 biți.

Registrele PITCNT0÷3 (PIT Count Register), 034Ah+4*x, x=0÷3.

Registrele $PITCNT_{0÷3}$ de câte 16 biți corespund temporizatoarelor de 16 biți (16-bit Timer) $0÷3$. Registrul $PITCNT_x$ este numărătorul temporizatorului x de 16 biți.

Perioada temporizării T_x pentru canalul $x=0÷3$ al modului PIT, utilizând temporizatorul $y=0÷1$ de 8 biți, este:

$$T_x = ((PITMTLD_y)+1)*((PITLD_x)+1)/f_{BUS}.$$

Aplicație

Se consideră o aplicație în care se utilizează canalele 0 și 1 ale modului PIT și temporizatorul 1 de 8 biți pentru a genera cereri de întrerupere cu frecvențele $f_0=1$ Hz și $f_1=4$ Hz. Să se calculeze valorile constantelor de timp care trebuie programate în registrele $PITMTLD_1$, $PITLD_0$ și $PITLD_1$. Se dă $f_{BUS}=8$ MHz.

$$\begin{aligned} ((PITMTLD_1)+1)*((PITLD_0)+1) &= f_{BUS}/f_0 = 8*10^6 \\ ((PITMTLD_1)+1)*((PITLD_1)+1) &= f_{BUS}/f_1 = 2*10^6 \end{aligned}$$

Se aleg valorile:

$$\begin{aligned} (PITMTLD_1)+1 &= 200 \\ (PITLD_0)+1 &= 40000 \\ (PITLD_1)+1 &= 10000 \end{aligned}$$

4.4. MODULUL TEMPORIZATOR CU FUNCȚII DE ACUMULARE, CAPTURĂ ȘI COMPARARE

Funcția de acumulare constă în numărarea unor evenimente definite prin fronturi ale unui semnal. Funcțiile de captură și comparare se referă la conținutul unui registru numărător comandat la intrare cu un semnal de frecvență dată. Funcția de captură constă în memorarea conținutului registrului numărător la un moment dat de un front al unui semnal. Funcția de comparare constă în generarea unui eveniment (front, cerere de întrerupere) în momentul în care conținutul registrului numărător ajunge la o anumită valoare. Rezultă că prin operații de captură se pot măsura parametri de timp ai unor semnale de intrare (durate ale impulsurilor) și prin operații de comparare se pot genera semnale cu parametri de timp stabiliți cu precizie (impulsuri cu durate de valori programabile cu precizie).

Modulul temporizator cu funcții de acumulare, captură și comparare **ECT** (*Enhanced Capture Timer*), fig. 4.5, conține patru canale cu funcții de acumulare, captură și comparare corespunzătoare pinilor P0÷3 și patru canale cu funcții de captură și comparare corespunzătoare pinilor P4÷7. Pentru implementarea funcțiilor de captură și comparare, modulul ECT conține un numărător principal de 16 biți (*16-Bit Free-Running Main Timer*) care numără impulsuri cu frecvența ciclurilor f_{BUS} (*Bus Clock*) divizată cu 1÷256 de circuitul prescalare (*Timer Prescaler*). Fiecare canal 0÷7 conține un registru de captură/comparare de 16 biți TC0÷7 (*Capture/Compare Register*) utilizat pentru memorarea conținutului registrului numărător principal în momentul capturii sau pentru scrierea constantei pentru comparare. De asemenea, fiecare canal 0÷7 conține un comparator (*Comparator*). Canalele 0÷3 conțin câte un registru de reținere de 16 biți TCxH, $x=0÷3$ (*Timer Input Capture Holding Register*) care permite salvarea unei valori captate în registrul de captură/comparare pentru o nouă operație de captare. Pentru funcția de acumulare, canalele 0÷3 conțin câte un registru acumulator (numărător) de impulsuri de 8 biți PAC0÷3 (*Pulse Accumulator Count Register*) și câte un registru de reținere de 8 biți PAXH, $x=0÷3$ (*Pulse Accumulator Holding Register*). Într-un registru PAXH se poate salva conținutul registrului PACx corespunzător pentru o nouă operație de numărare.

Modulul ECT conține și un temporizator format din circuitul prescalare (*Modulus Prescaler*), registrul pentru constanta de timp (*16-Bit Load Register*) și registrul numărător modulo cu decrementare (*16-Bit Modulus Down Counter*), figura 4.5. Acest temporizator poate fi utilizat independent pentru generarea periodică de întreruperi sau pentru comanda canalelor 0÷3 de captură și acumulare. Astfel, fiecare moment de temporizare (trecere prin zero a conținutului registrului numărător modulo cu decrementare) poate comanda transferul în registrele de reținere pentru canalele 0÷3 de captură și acumulare.

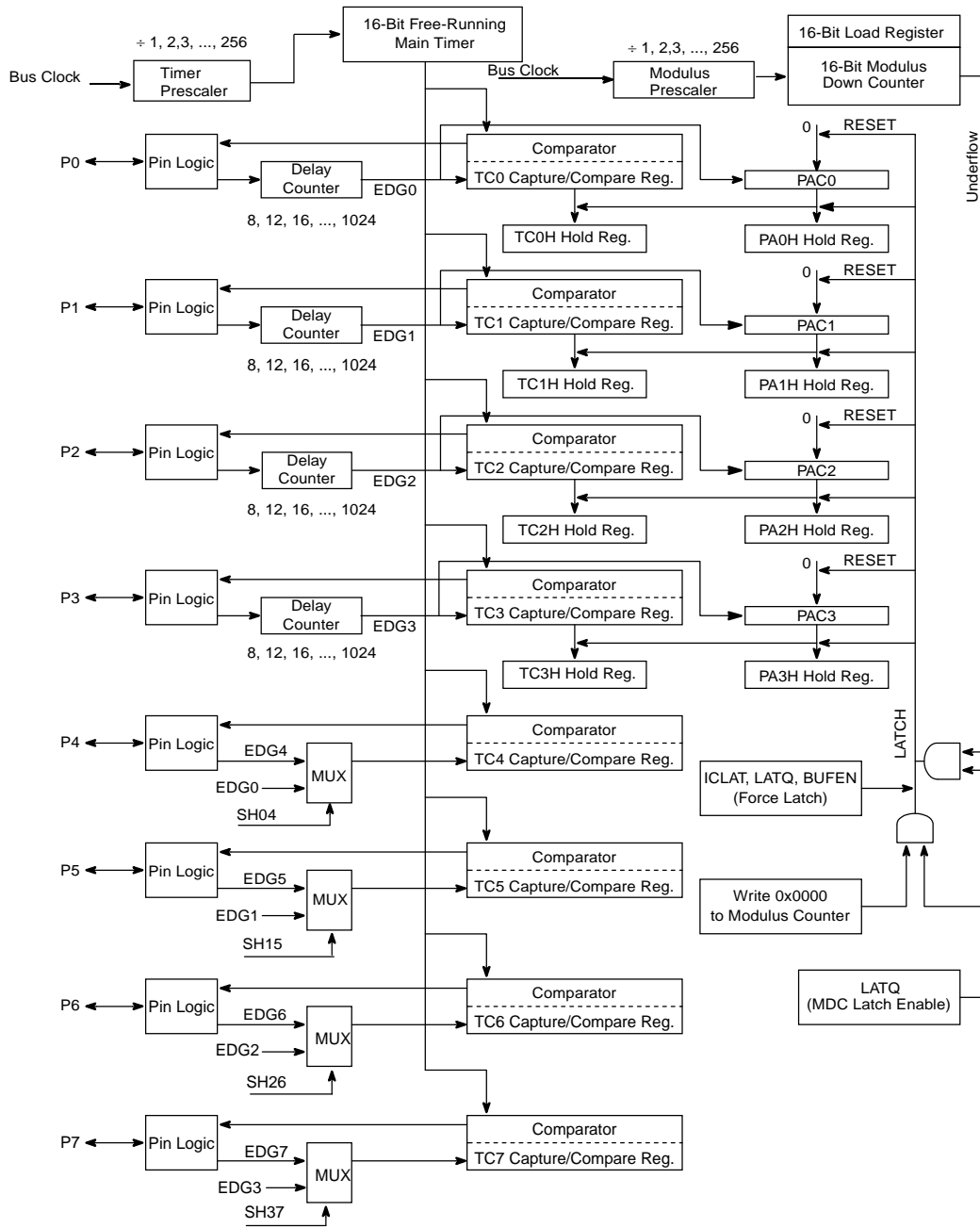


Fig. 4.5. Structura modului temporizator cu funcții de acumulare, captură și comparare.

Pentru funcțiile de captură și acumulare fiecare canal al modulului ECT conține câte un circuit detector de front (*Edge Detector*) care poate fi programat pentru detecția fronturilor de ridicare sau/și coborâre. Canalele 0÷3 conțin câte un circuit de întârziere (*Delay Counter*) pentru rejectia impulsurile de intrare cu durată mai mică decât durata întârzierii, impulsuri considerate perturbații. Durata întârzierii pentru un circuit se stabilește prin contorizarea unui număr programabil de impulsuri cu frecvența ciclurilor f_{BUS} (*Bus Clock*).

Aplicație

La pinul P0 corespunzător canalului 0 al modulului ECT se aplică un impuls pentru determinarea duratei acestuia utilizând funcția de captură. Detectorul de front este programat pentru detecție pe front de ridicare și pe front de coborâre. Valorile captate în registrul de captură/comparare TC0 și în registrul de reținere TC0H sunt 50000 și respectiv 10000. Se consideră că între cele două fronturi ale impulsului nu s-a produs depășire la numărătorul principal. Circuitul prescalare al numărătorului principal divizează frecvența de tact f_{BUS} (*Bus Clock*) cu $d=100$. Se dă $f_{BUS}=8$ MHz.

Durata t_i a impulsului este:

$$t_i = ((TC0) - (TC0H)) / (f_{BUS}/d) = 0,5 \text{ s.}$$

Aplicație

Se pune problema de a genera la pinul P0 corespunzător canalului 0 al modulului ECT a unor impulsuri de frecvență și durată date. În acest scop, se utilizează funcția de comparare pentru canalul 0 cu comutarea nivelului logic al pinului în momentele de comparare (conținutul registrului numărător principal coincide cu conținutul registrului de captură/comparare TC0) precum și în momentele de depășire la numărătorul principal.

Se cere să se calculeze factorul de divizare d corespunzător circuitului prescalare al numărătorului principal și valoarea constantei pentru comparare (TC0) pentru generarea de impulsuri cu frecvența $f_i=12,5$ Hz și durata $t_i=20$ ms. Se dă $f_{BUS}=8$ MHz.

Factorul de divizare d este

$$d = f_{BUS} / (2^{16} * f_i) \cong 10.$$

Constanta pentru comparare se obține din relația

$$t_i = (2^{16} - (TC0)) / (f_{BUS}/d).$$

Rezultă

$$(TC0) \cong 3 * 2^{14}.$$

4.5. MODULUL GENERATOR DE IMPULSURI MODULATE ÎN DURATĂ

Modulul generator de impulsuri modulate în durată PWM (*Pulse-Width Modulator*) conține 8 canale care generează la pinii PWM0÷7 (portul P, PP0÷7) ai microcontrolerului impulsuri cu durate și perioade programabile, stabilite prin numărare de impulsuri de frecvență dată.

Principalele componente ale unui canal $x=0\div7$ al modulului PWM, fig. 4.6, sunt:

- registru numărător de 8 biți PWMCNT x (*PWM Channel Counter Register*);
- registru pentru perioadă PWMPER x (*PWM Channel Period Register*);
- registru pentru durată PWMDTY x (*PWM Channel Duty Register*);
- două comparatoare de câte 8 biți pentru perioadă și durată (*8-bit Compare =*).

Validarea unui canal PWM cu bitul PWME x din registrul de validare PWME (*PWM Enable Register*) determină accesul semnalului de tact (*Clock Source*) la intrarea numărătorului PWMCNT x și conectarea ieșirii canalului PWM la pinul PP x prin multiplexorul de la ieșire. Fronturile corespunzătoare impulsurilor se generează de circuitele comparatoare la momentele de egalitate între conținuturile registrelor PWMCNT x și PWMPER x pentru perioadă și între conținuturile registrelor PWMCNT x și PWMDTY x pentru durată.

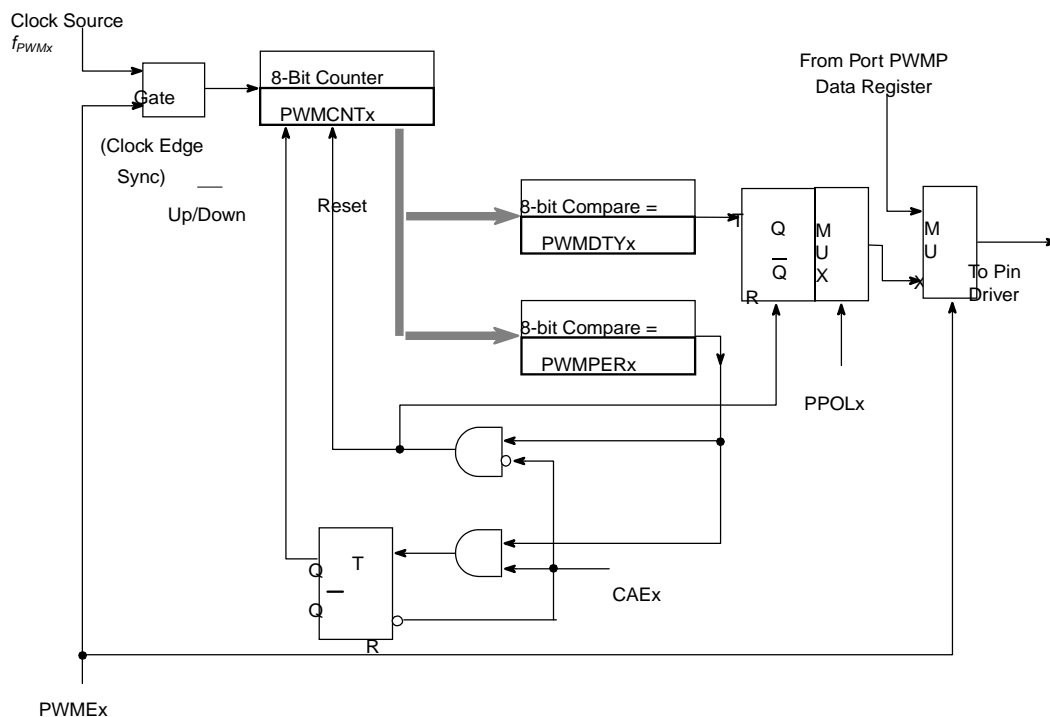


Fig. 4.6. Structura unui canal al modulului generator de impulsuri modulate în durată.

Generarea impulsurilor modulate în durată de un canal al modului PWM se realizează în una dintre variantele cu aliniere la stânga sau cu aliniere centrală, conform programării cu bitul CAEx din registrul de validare aliniere centrală PWMCAE (*PWM Center Align Enable Register*).

Logica de funcționare a unui generator de impulsuri modulate în durată cu aliniere la stânga este prezentată în figura 4.7 și este implementată cu circuitele din figura 4.6. În această variantă, registrul PWMCNTx numără impulsuri de tact cu frecvența f_{PWMx} în sens crescător conform sistemului modulo (PWMPERx). În fiecare moment de egalitate între conținuturile registrelor PWMCNTx și PWMDTYx, circuitul comparator corespunzător generează un impuls care comandă prin intrarea T a bistabilului de ieșire comutarea la nivel logic 1 a ieșirii Q a acestuia. În momentele de trecere pe zero ale conținutului registrului PWMCNTx se comandă comutarea Q=0 a bistabilului de ieșire prin intrarea R. Durata și perioada impulsurilor generate sunt indicate în fig. 4.7. Multiplexorul comandat cu bitul PPOLx al registrului polaritate PWMPOL (*PWM Polarity Register*) selectează ieșirea Q sau /Q a bistabilului de ieșire pentru stabilirea polarității impulsurilor generate.

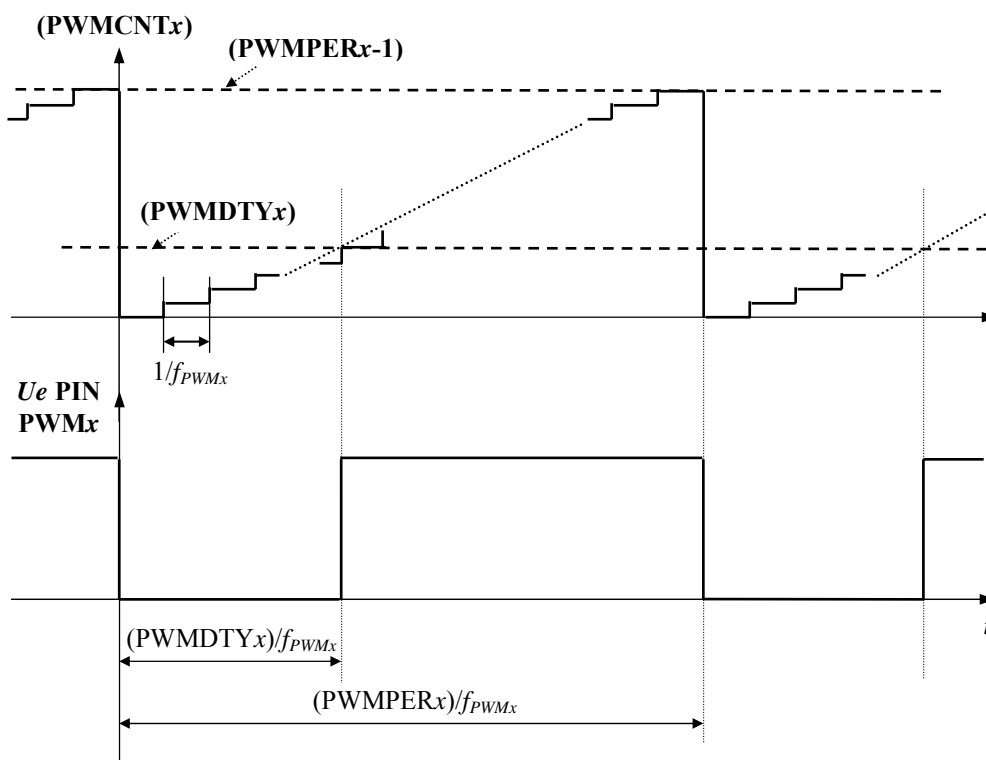


Fig. 4.7. Logica de funcționare a unui generator de impulsuri modulate în durată cu aliniere la stânga.

Logica de funcționare a unui generator de impulsuri modulate în durată cu aliniere centrală este prezentată în figura 4.8 și este implementată cu circuitele din figura 4.6. În această variantă, registrul PWMCNT_x numără impulsuri de tact cu frecvența f_{PWMx} în sens crescător de la zero până la valoarea dată de conținutul registrului PWMPER_x, după care numără în sens descrescător până la anularea conținutul registrului PWMCNT_x. În fiecare moment de egalitate între conținuturile registrelor PWMCNT_x și PWMDTY_x, circuitul comparator corespunzător generează un impuls care comandă prin intrarea T a bistabilului de ieșire comutarea acestuia. Durata și perioada impulsurilor generate sunt indicate în figura 4.8. Multiplexorul comandat cu bitul PPOL_x al registrului polaritate PWMPOL (*PWM Polarity Register*) selectează ieșirea Q sau /Q a bistabilului de ieșire pentru stabilirea polarității impulsurilor generate.

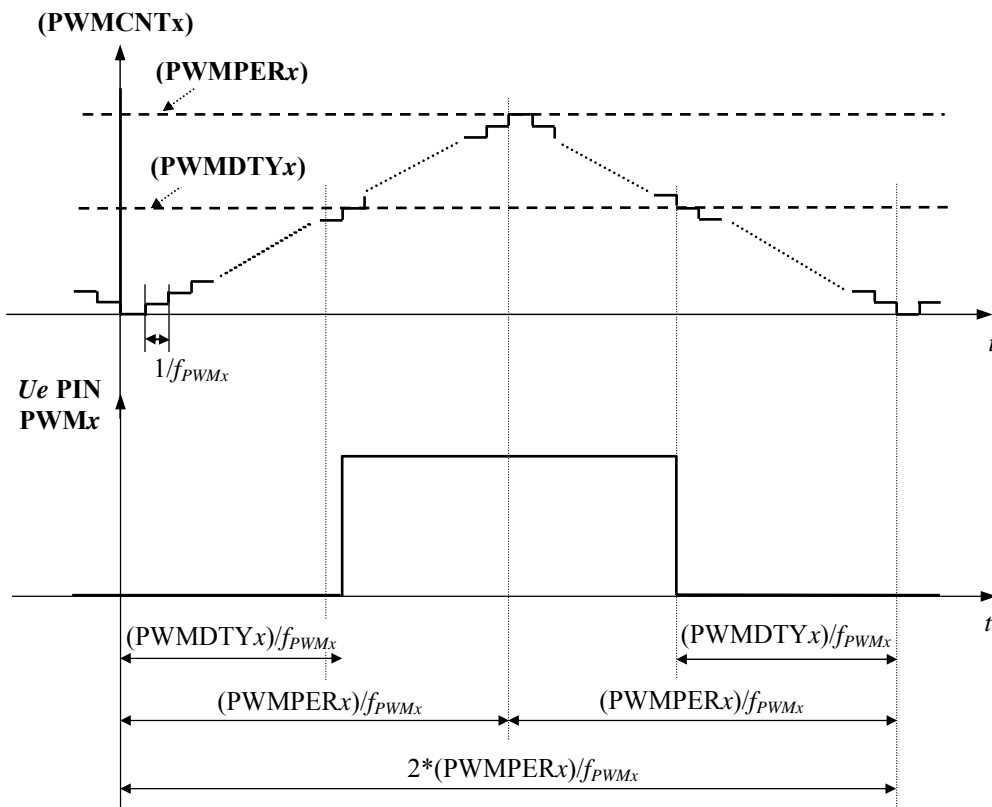


Fig.4.8. Logica de funcționare a unui generator de impulsuri modulate în durată cu aliniere centrală.

În ambele variante prezentate mai sus, factorul de umplere corespunzător impulsurilor generate este funcție de raportul între conținutul registrului pentru durată

PWMDTY_x și conținutul registrului pentru perioadă PWMPER_x. Frecvența impulsurilor generate este $f_{PWMx}/(PWMPERx)$ în cazul variantei cu aliniere la stânga și $f_{PWMx}/(2*(PWMPERx))$ în cazul variantei cu aliniere centrală.

Modulul PWM conține circuite de programare și selecție a frecvențelor de tact f_{PWMx} pentru cele 8 canale generatoare de impulsuri modulate în durată, figura 4.9. Frecvențele f_{PWMx} se obțin din frecvența ciclurilor f_{BUS} (*Bus Clock*) prin divizare. După o primă divizare cu circuitul prescalare (*Prescaler*) se obțin semnale cu frecvențele f_{BUS}/d , $d=1, 2, 4, 8, 16, 32, 64$ și 128 , conectate la intrările celor două multiplexoare cu câte 8 intrări, fig. 4.9. Prin comanda multiplexoarelor cu biții registrului PWMPRCLK (*PWM Prescaler Clock Select Register*) se obțin semnalele de tact *Clock A* și *Clock B*. Fiecare din aceste două semnale este divizat cu câte un numărator de 8 biți cu decrementare modulo p_A , respectiv p_B , unde p_A este conținutul registrului de scalare PWMSCLA (*PWM Scale A Register*) și p_B este conținutul registrului de scalare PWMSCLB (*PWM Scale B Register*). Prin divizarea cu 2 a frecvențelor semnalelor de la ieșirile celor două număratoare modulo cu decrementare, se obțin semnalele de tact *Clock SA* și *Clock SB*. Semnalele cu frecvențele de tact f_{PWMx} pentru cele 8 canale generatoare de impulsuri modulate în durată se selectează din variantele *Clock A*, *Clock SA* și *Clock B*, *Clock SB* prin comanda celor opt multiplexoare cu câte 2 intrări cu biții registrului PWMCLK (*PWM Clock Select Register*).

Aplicație

Să se precizeze modalitățile prin care la ieșirile canalelor 0 și 1 ale modulului PWM se generează impulsuri de frecvențe minimă și respectiv maximă. Să se calculeze aceste frecvențe, dacă frecvența ciclurilor este $f_{BUS}=8$ MHz.

Generarea de impulsuri de frecvență minimă se obține prin:

- selecția semnalului de frecvență $f_{BUS}/128$ de la circuitul prescalare;
- programarea număratorului modulo p_A cu decrementare pentru divizare de frecvență cu 256, $p_A=(PWMSCLA)=0$;
- selecția semnalului *Clock SA* pentru comanda canalului 0 cu frecvența de tact $f_{PWM0}=f_{BUS}/(128*256*2)=f_{BUS}/2^{16}$;
- programarea generării de impulsuri modulate în durată cu aliniere centrală și cu perioadă maximă, $(PWMPER0)=255$.

Rezultă frecvența minimă $f_{min}=f_{PWM0}/(2*255)\cong f_{BUS}/2^{25}\cong 0,24$ Hz.

Generarea de impulsuri de frecvență maximă se obține prin:

- selecția semnalului de frecvență f_{BUS} de la circuitul prescalare;
- selecția semnalului *Clock A* pentru comanda canalului 1 cu frecvența de tact $f_{PWM0}=f_{BUS}$;
- programarea generării de impulsuri modulate în durată cu aliniere la stânga, cu perioadă și durată minime, $(PWMPER1)=2$ și $(PWMDTY1)=1$.

Rezultă frecvența maximă $f_{max}=f_{PWM0}/2=f_{BUS}/2=4$ MHz.

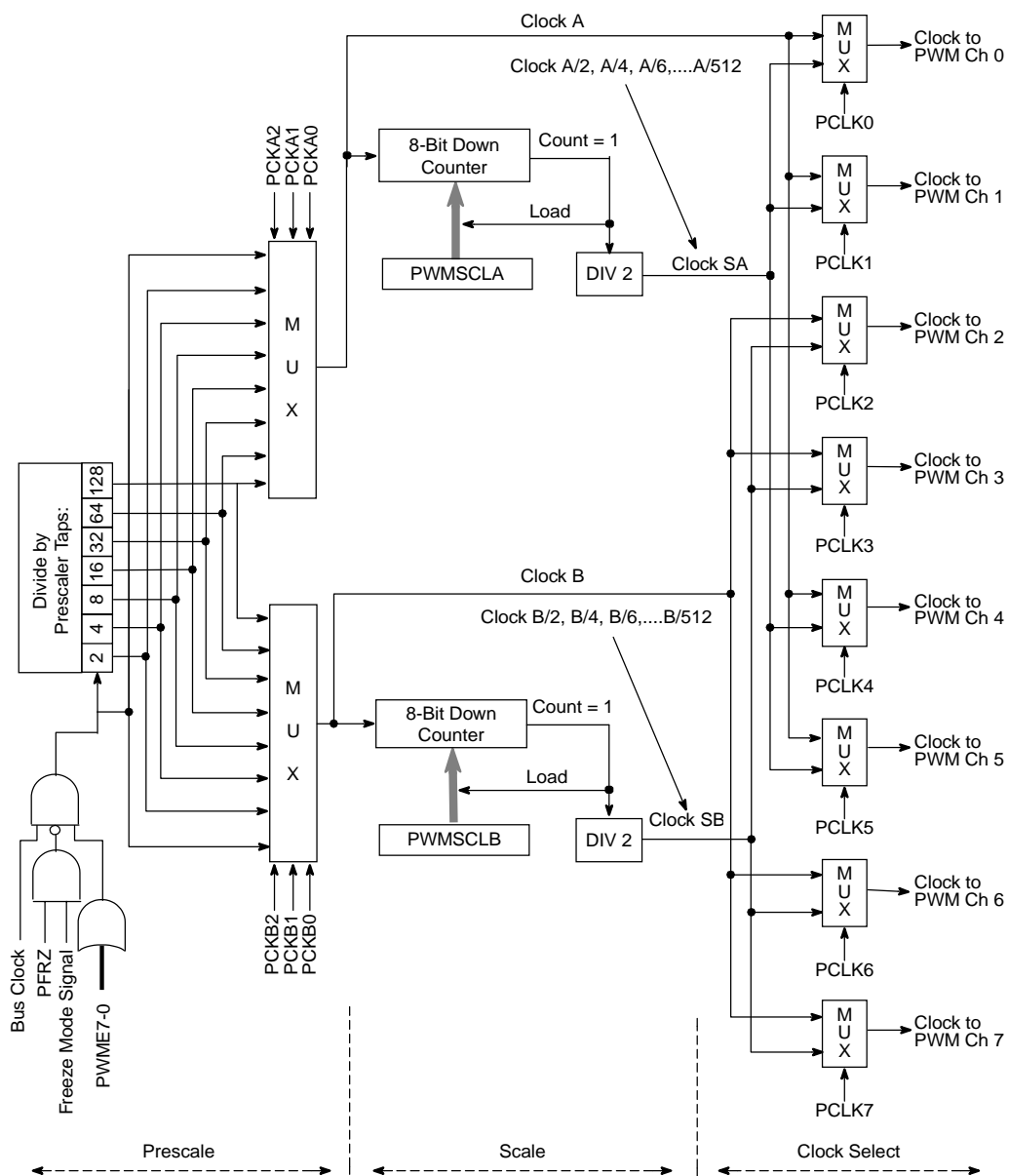


Fig. 4.9. Structura circuitelor de programare și selecție a frecvențelor de tact pentru canalele modului generator de impulsuri modulate în durată.

Aplicație

Se definește factorul de umplere $F_u = t_H / T$, unde t_H este durata nivelului 1 (ridicat) al impulsurilor și $T = 1/f$ este perioada impulsurilor.

Se consideră o aplicație în care se generează impulsuri cu frecvența $f_0 = 10$ kHz, durata $t_{H0} = 20$ μ s la canalul 0 al modulului PWM și impulsuri cu frecvența $f_1 = 1$ kHz, durata de $t_{H1} = 200$ μ s la canalul 1 al modulului PWM.

- deoarece $f_0 > f_1$, se alege semnalul $f_{Clock A}$ pentru canalul 0 și semnalul $f_{Clock SA}$ pentru canalul 1;

- se alege $f_{PWM0}/f_{PWM1} = f_{Clock A}/f_{Clock SA} = f_0/f_1 = 10$;

- rezultă $(PWMSCLA) = 5$;

- din relația $f_0 = f_{BUS}/(d*(PWMPER0))$, rezultă $d*(PWMPER0) = 800$;

- se alege $d = 8$ și $(PWMPER0) = (PWMPER1) = 100$;

- se alege ca impulsurile să înceapă cu nivel logic 1 prin programarea biților $PPOL1 = 0$ la nivel logic 1;

- deoarece factorul de umplere al impulsurilor pentru cele două canale 0 și 1 ale modulului PWM este $1/5$, rezultă $(PWMDTY0) = (PWMDTY1) = 20$.

4.6. MODULUL DE CONVERSIE ANALOG NUMERICĂ

Un microcontroler din familia HCS12X conține două module de conversie analog numerică ATD0 și ATD1 (*Analog-to-Digital Converter*) de câte 10 biți care au structuri similare și care diferă prin numărul de canale analogice de intrare 8 și respectiv 16. În cele ce urmează se prezintă modulul ATD0, figura 4.10.

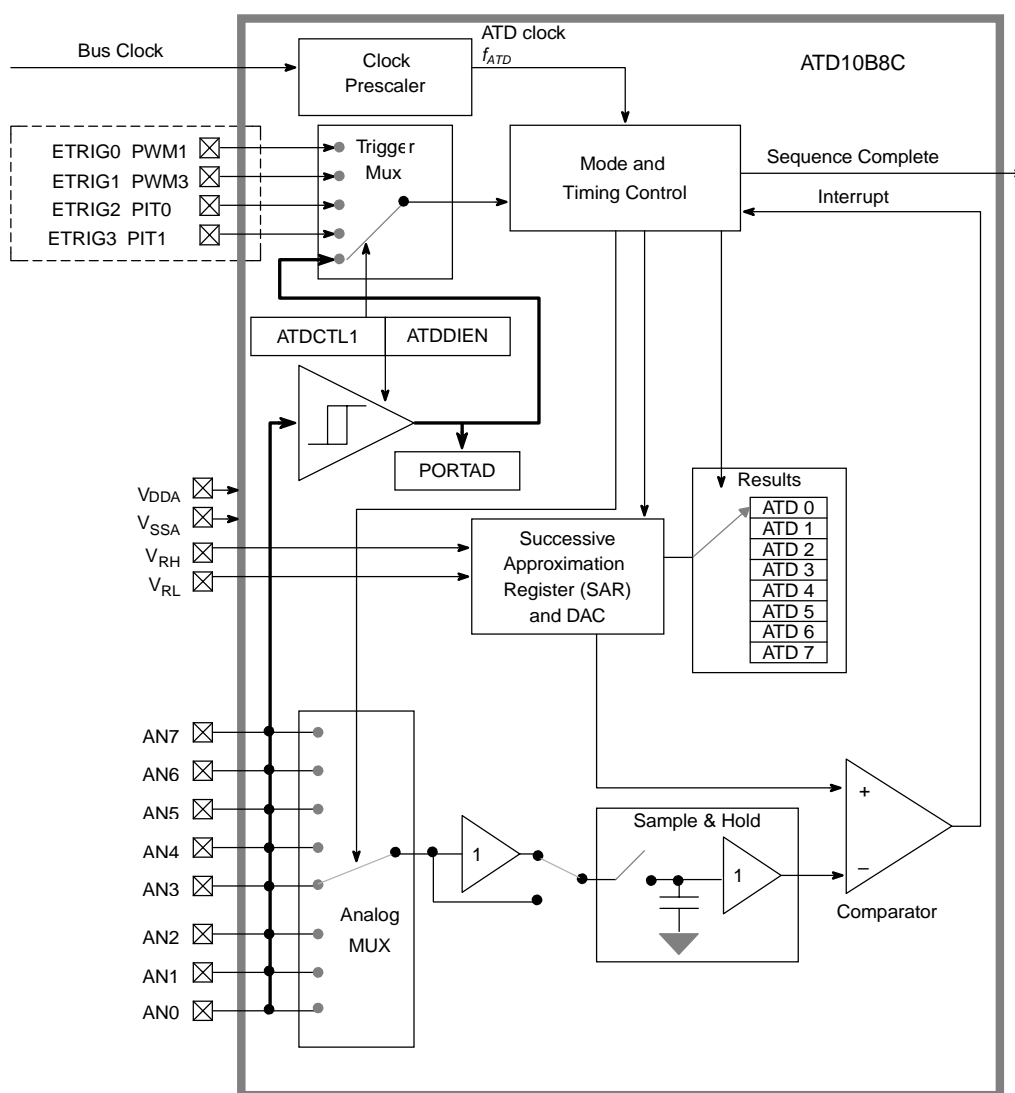


Fig. 4.10. Structura modulului de conversie analog numerică ATD0.

Frecvența de tact conversie f_{ATD} (*ATD Clock*) cu valori între 0,5÷2 MHz se obține din frecvența ciclurilor de magistrală f_{BUS} prin divizare de frecvență în circuitul prescalare (*Clock Prescaler*). Se obține $f_{ATD} = f_{BUS} / (2 * (PRS + 1))$, unde PRS este valoarea factorului de prescalare dat prin 5 biți $PRS4 \div 0$ din registrul ATDCTL4 (*ATD Control Register 4*).

Pentru a obține valoarea numerică de la un moment dat a semnalului analogic conectat la un pin AN_x , $x=0 \div 7$ sunt necesare operațiile:

- comanda multiplexorului (*Analog MUX*);
- eșantionarea semnalului analogic cu circuitul de eșantionare și memorare (*Sample & Hold*);
- conversia analog numerică de tipul cu aproximații succesive.

Eșantionarea semnalului analogic se realizează în două faze. În prima fază, semnalul analogic este conectat prin amplificatorul repetor la intrarea circuitului de eșantionare și memorare, pe durata a două perioade ale semnalului de tact f_{ATD} , pentru încărcarea rapidă a condensatorului de memorare, fig. 4.10. În faza a doua, semnalul analogic este conectat direct la intrarea circuitului de eșantionare și memorare, pentru încărcarea cu precizie a condensatorului de memorare. Durata celei de-a doua faze este de 2, 4, 8 sau 16 perioade a semnalului de tact f_{ATD} și se stabilește prin programare cu biții $SMP1 \div 0$ din registrul ATDCTL4 (*ATD Control Register 4*). Pe durata conversiei analog numerice circuitul de eșantionare și memorare se comandă în stare de memorare.

Pentru conversia analog numerică cu aproximații succesive se utilizează:

- circuitul comparator (*Comparator*);
- registrul de aproximații succesive (*SAR*);
- convertorul numeric analogic de 10 biți (*DAC*);
- logica de control (*Mode and Timing Control*).

O conversie analog numerică se realizează pe $N=8$ sau $N=10$ biți, funcție de nivelul logic al bitului $SRES8$ din registrul ATDCTL4. Durata unei conversii analog numerice este de N perioade ale semnalului de tact f_{ATD} . Rezultatul unei conversii analog numerice se scrie în unul dintre registrele de câte 16 biți ATD_x , $x=0 \div 7$, compus din registrele de câte 8 biți $ATDDR_xH$ și $ATDDR_xL$ (*ATD Result Register*). Această scriere se poate face prin aliniere la stânga (*Left Justified*) sau prin aliniere la dreapta (*Right Justified*), funcție de nivelul logic al bitului DJM din registrul ATDCTL5. De asemenea, rezultatul unei conversii analog numerice se obține în reprezentare cu semn sau fără semn, funcție de nivelul logic al bitului $DSGN$ din registrul ATDCTL5. În tabelul 4.2 se indică variantele de reprezentare a rezultatelor conversiilor analog numerice și pozițiile ocupate în registrele ATD_x , funcție de biții $SRES8$, DJM și $DSGN$. În tabelul 4.3 se indică rezultatele conversiilor analog numerice pe 8 și 10 biți, pentru diferite valori ale tensiunii de intrare, prin alinierea rezultatelor la stânga, în reprezentare cu semn și fără semn. Rezultatele aliniate la stânga a conversiilor analog numerice pe 8 biți se scriu în registrele $ATDDR_xH$, iar rezultatele aliniate la stânga a conversiilor analog numerice pe 10 biți se scriu în

registrele ATDDR x H și în cei doi biți mai semnificativi ai registrelor ATDDR x L, ceea ce rezultă și din tabelul 4.2.

Tabelul 4.2

SRES8	DJM	DSGN	Result Data Formats Description and Bus Bit Mapping
1	0	0	8-bit / left justified / unsigned — bits 8–15
1	0	1	8-bit / left justified / signed — bits 8–15
1	1	X	8-bit / right justified / unsigned — bits 0–7
0	0	0	10-bit / left justified / unsigned — bits 6–15
0	0	1	10-bit / left justified / signed — bits 6–15
0	1	X	10-bit / right justified / unsigned — bits 0–9

Tabelul 4.3

Input Signal $V_{RL} = 0$ Volts $V_{RH} = 5.12$ Volts	Signed 8-Bit Codes	Unsigned 8-Bit Codes	Signed 10-Bit Codes	Unsigned 10-Bit Codes
5.120 Volts	7F	FF	7FC0	FFC0
5.100	7F	FF	7F00	FF00
5.080	7E	FE	7E00	FE00
2.580	01	81	0100	8100
2.560	00	80	0000	8000
2.540	FF	7F	FF00	7F00
0.020	81	01	8100	0100
0.000	80	00	8000	0000

Declanșarea modului ATD se referă la secvențe de conversii analog numerice (*conversion sequence*). O secvență de conversii poate cuprinde mai multe conversii analog numerice, cel mult 8. Numărul de conversii dintr-o secvență se stabilește cu biții S8C, S4C, S2C și S1C din registrul ATDCTL3. Conversiile analog numerice dintr-o secvență corespund unui singur canal analogic de intrare AN x selectat cu biții CC, CB și CA din registrul ATDCTL5 sau pentru mai multe canale succesive începând cu cel selectat cu biții CC, CB și CA, în funcție de nivelul logic programat în bitul MULT (*Multi-Channel Sample Mode*) din registrul ATDCTL5. Canalele succesive se obțin prin incrementare cu câte o unitate în sistemul modulo $y+1$, $y=1\div 7$, unde y este o valoare programată în biții WRAP2 \div 0 din registrul ATDCTL0.

Declanșarea secvențelor de conversii se poate realiza prin scriere în registrul ATDCTL5, cu semnal extern conectat la un pin AN0 \div 7 sau cu un semnal extern modului ATD prin conexiunile ETRIG0 \div 3, figura 4.10. Prin conexiunile ETRIG0 \div 3,

declanșarea secvențelor de conversii se poate comanda de la canalele 1 sau 3 ale modulului PWM sau de la canalele 0 sau 1 ale modulului PIT, figura 4.10. Selecția unui pin AN0÷7 sau a unei conexiuni ETRIG0÷3, pentru declanșare externă, se realizează cu biții ETRIGSEL, ETRIGCH2÷0 din registrul ATDCTL1 care comandă multiplexorul pentru declanșare (*Trigger Mux*). În cazul utilizării unui canal AN_x pentru declanșare externă, este necesară validarea ieșirii corespunzătoare a circuitului *digital input buffer*. Cele 8 ieșiri ale acestui circuit pot fi validate individual cu biții registrului ATDDIEN (*ATD Input Enable Register*).

Variantele de declanșare a secvențelor de conversii, funcție de biții ETRIGLE, ETRIGP, ETRIGE din registrul ATDCLT2 și de bitul SCAN din registrul ATDCTL5, sunt indicate în tabelul 4.5.

Tabelul 4.5

ETRIGLE	ETRIGP	ETRIGE	SCAN	Description
X	X	0	0	Ignores external trigger. Performs one conversion sequence and stops.
X	X	0	1	Ignores external trigger. Performs continuous conversion sequences.
0	0	1	X	Falling edge triggered. Performs one conversion sequence per trigger.
0	1	1	X	Rising edge triggered. Performs one conversion sequence per trigger.
1	0	1	X	Trigger active low. Performs continuous conversions while trigger is active.
1	1	1	X	Trigger active high. Performs continuous conversions while trigger is active.

Momentul terminării unei conversii analog numerice dintr-o secvență de conversii pentru un canal AN_x, concretizată prin încărcarea rezultatului acesteia în registrul ATD_x este indicat prin poziționarea la nivel logic 1 a bitului CCF_x (*Conversion Complete Flag x*) din registrul ATDSTAT1 (*ATD Status Register 1*).

În momentul terminării tuturor conversiilor analog numerice dintr-o secvență se generează o cerere de întrerupere.

Aplicație

Să se determine valorile posibile ale factorului de prescalare PRS , pentru $f_{BUS}=8$ MHz.

Frecvența de tact conversie f_{ATD} (*ATD Clock*) cu valori între 0,5÷2 MHz se obține din frecvența ciclurilor de magistrală f_{BUS} prin divizare de frecvență în circuitul prescalare (*Clock Prescaler*). Se obține $f_{ATD} = f_{BUS} / (2 * (PRS + 1))$, unde PRS este valoarea factorului de prescalare dat prin 5 biți $PRS4 \div 0$ din registrul $ATDCTL4$ (*ATD Control Register 4*).

- se notează $f_{ATDmax}=2$ MHz și $f_{ATDmin}=0,5$ MHz
- valorile posibile ale factorului de prescalare PRS rezultă din relația:

$$f_{BUS} / (2 * f_{ATDmax}) - 1 \leq PRS \leq f_{BUS} / (2 * f_{ATDmin}) - 1$$

$$2 \leq PRS \leq 8$$

Aplicație

Să se determine durata minimă a unei conversii analog numerice, T_{min} .

Pentru realizarea unei conversii analog numerice sunt necesare:

- 2 perioade f_{ATD} (ale semnalului de tact conversie *ATD Clock*) pentru prima fază de eșantionare;
- minim 2 perioade f_{ATD} pentru a doua fază de eșantionare;
- N perioade f_{ATD} pentru conversia analog numerică cu aproximații succesive pe $N=8$ sau 10 biți;
- se alege $f_{ATDmax}=2$ MHz;
- rezultă $T_{min} = (N+4) / f_{ATDmax}$, $T_{min}=6$ μs, pentru $N=8$ și $T_{min}=7$ μs, pentru $N=10$.

4.7. MODULUL DE INTERFAȚĂ PENTRU COMUNICAȚIE SERIALĂ ASINCRONĂ

Comunicația serială între un echipament emițător și un echipament receptor se realizează printr-o singură linie. Rezultă că, la un moment dat, se poate transfera un singur bit între cele două echipamente. Transferul serial al unui cuvânt (octet) de la emițător la receptor se realizează prin multiplexare, respectiv demultiplexare. Rezultă că la emițător se efectuează conversia paralel-serie a cuvintelor, iar la receptor se efectuează conversia serie-paralel a cuvintelor.

Pentru sincronizarea transferului de date în comunicația serială asincronă nu se utilizează un semnal de tact comun pentru emițător și receptor. Pentru sincronizarea transferului de date, fiecare cuvânt transmis este încadrat între un bit de start și un bit de stop, figura 4.11. Un cuvânt cuprinde o secvență de 8 sau 9 biți de date. Transmisia serială a unui cuvânt de către emițător constă în încărcarea succesivă a liniei de comunicație cu biții corespunzători cuvântului, cu frecvența de comunicație f_c , exprimată în biți/s. Recepția serială constă în testarea periodică a stării liniei de comunicație în scopul determinării secvenței de biți încărcată de emițător. Pentru transferul corect al unui cuvânt este necesar ca receptorul să eșantioneze starea liniei de comunicație cu aceeași frecvență și fază cu care se încarcă biții cuvântului de către emițător. În acest scop, comunicația serială asincronă se realizează cu o frecvență de comunicație f_c stabilită (programată) pentru oscilatorul de la emisie și pentru oscilatorul de la recepție. Sincronizarea de fază a oscilatorului de la recepție cu cel de la emisie se produce pe baza frontului de cădere corespunzător bitului de start care precede biții de date în procesul de transmisie a unui cuvânt. După bitul de start cu nivel logic 0 urmează emisia biților de date (8 sau 9) începând cu bitul de semnificație minimă. Ultimul bit de date emis poate fi bit de paritate utilizat la recepție pentru verificare de erori. După ultimul bit de date urmează emisia unui bit de stop cu nivel logic 1.

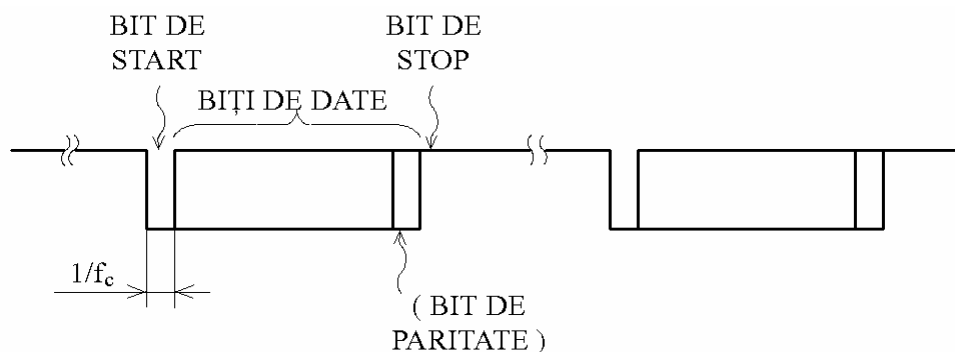


Fig. 4.11. Structura cuvintelor în comunicația serială asincronă.

Comunicația serială asincronă între două echipamente se poate realiza în ambele sensuri alternativ, printr-o singură linie, în cazul comunicației semiduplex (*half-duplex*) sau simultan, prin două linii, în cazul comunicației duplex (*full-duplex*).

Modulul de interfață pentru comunicație serială asincronă **SCI** (*Serial Communication Interface*) al unui microcontroler din familia HCS12X conține 6 canale duplex cu structuri identice, figura 4.12, care pot fi utilizate pentru comunicația cu echipamente periferice externe. Prezentarea care urmează se referă la un singur canal al modului SCI.

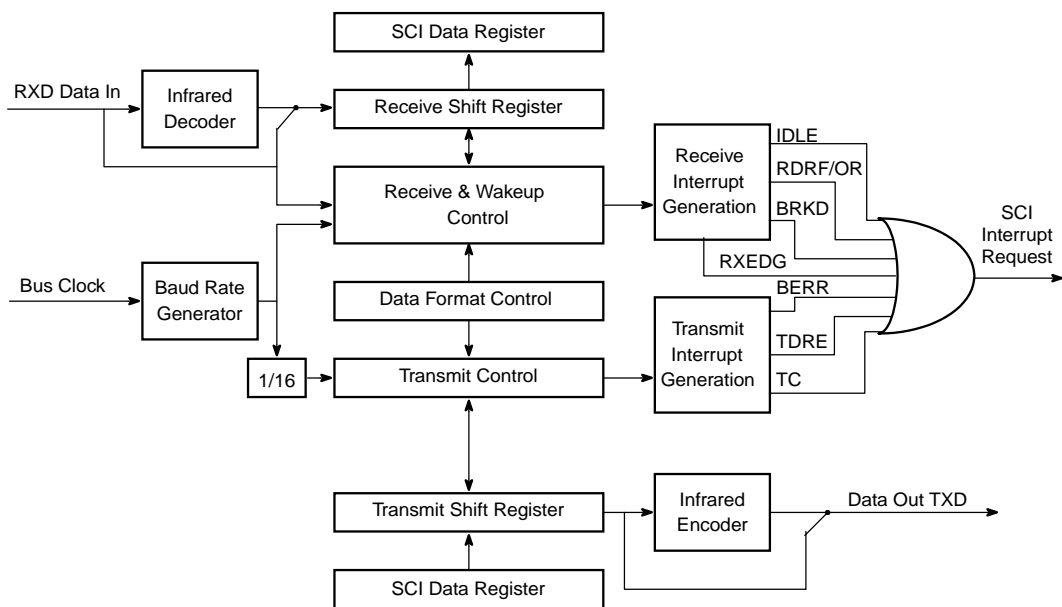


Fig. 4.12. Structura unui canal al modului de interfață pentru comunicație serială asincronă.

Conexiunile la liniile de comunicație serială se realizează prin pinul de recepție date (*RXD Data In*) și prin pinul de transmisie date (*TXD Data Out*). Pentru comunicație serială asincronă în infraroșu, canalul modului SCI conține un circuit decodor infraroșu (*Infrared Decoder*), la recepție și un circuit codor infraroșu, la transmisie care realizează conversia datelor între cele două reprezentări indicate în figura 4.13. Din această figură rezultă că reprezentarea datelor pentru comunicație serială asincronă în infraroșu se realizează prin impulsuri înguste (cu durate mai mici decât $1/f_c$), câte un impuls pentru fiecare bit cu nivel logic zero. Pentru conversia serie-paralel se utilizează registrul de deplasare recepție (*Receive Shift Register*), iar pentru conversia paralel-serie se utilizează registrul de deplasare transmisie (*Transmit Shift Register*).

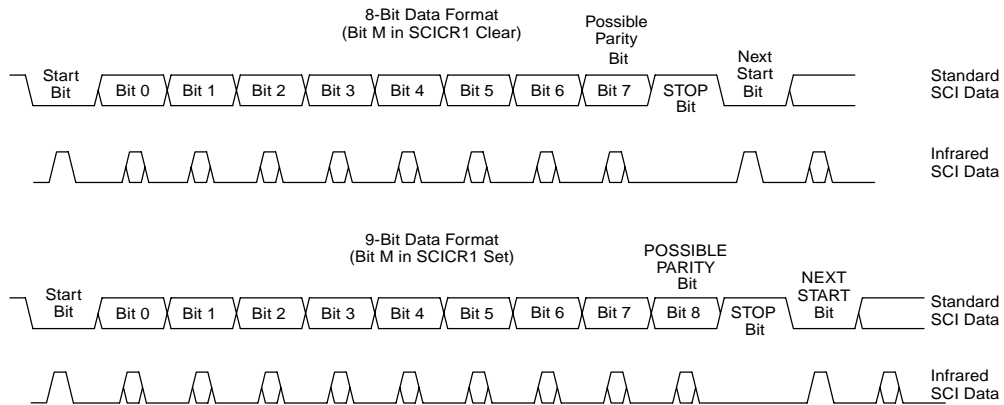


Fig. 4.13. Reprezentarea datelor pentru comunicație serială asincronă.

Canalul SCI conține două registre de date (*SCI Data Register*), unul pentru recepție și unul pentru transmisie, cărora le corespunde aceeași adresă. Un cuvânt asamblat prin conversie serie-paralel în registrul de deplasare recepție este transferat în registrul de date recepție care poate fi accesat prin citire. Registrul de date transmisie este accesat prin scriere a unui cuvânt de transmis. Cuvântul scris în registrul de date transmisie este transferat în registrul de deplasare transmisie pentru conversie paralel-serie.

Frecvența de comunicație f_c se obține din frecvența ciclurilor f_{BUS} prin divizare în circuitul (*Baud Rate Generator*) cu un factor de divizare cu valori în intervalul 1÷8191, dat de biții $SCIBR.12\div 0$ din registrul *SCIBD* (*SCI Baud Rate Register*). Deoarece se realizează încă o divizare de frecvență cu 16, se obține:

$$f_c = f_{BUS} / (16 * (SCIBR.12 \div 0)).$$

Canalul SCI conține circuite logice pentru controlul și activarea recepției (*Receive & Wakeup Control*) și circuite logice pentru controlul transmisiei (*Transmit Control*), figura 4.12. Sincronizarea funcționării canalului SCI cu unitatea centrală de prelucrare se poate realiza prin cereri de întrerupere.

Aplicație

Să se calculeze factorul de divizare dat de biții $SCIBR.12\div 0$ pentru o comunicație serială asincronă cu frecvența $f_c = 9600$ Hz (biți/secundă). Se dă $f_{BUS} = 8$ MHz.

$$(SCIBR.12 \div 0) = f_{BUS} / (16 * f_c) = 52,08$$

Pentru funcționarea și programarea unui canal SCI se utilizează registre ale procesorului, dintre care se indică:

- registrul SCIBD (*SCI Baud Rate Register*);
- registrul SCICR1 (*SCI Control Register 1*);
- registrul SCICR2 (*SCI Control Register 2*);
- registrul SCISR1 (*SCI Status Register 1*);
- registrul SCISR2 (*SCI Status Register 2*);
- registrele SCIDRH, SCIDRL (*SCI Data Registers*).

În cele ce urmează se face o prezentare parțială a structurii și funcționării circuitelor de transmisie ale unui canal SCI, figura 4.14.

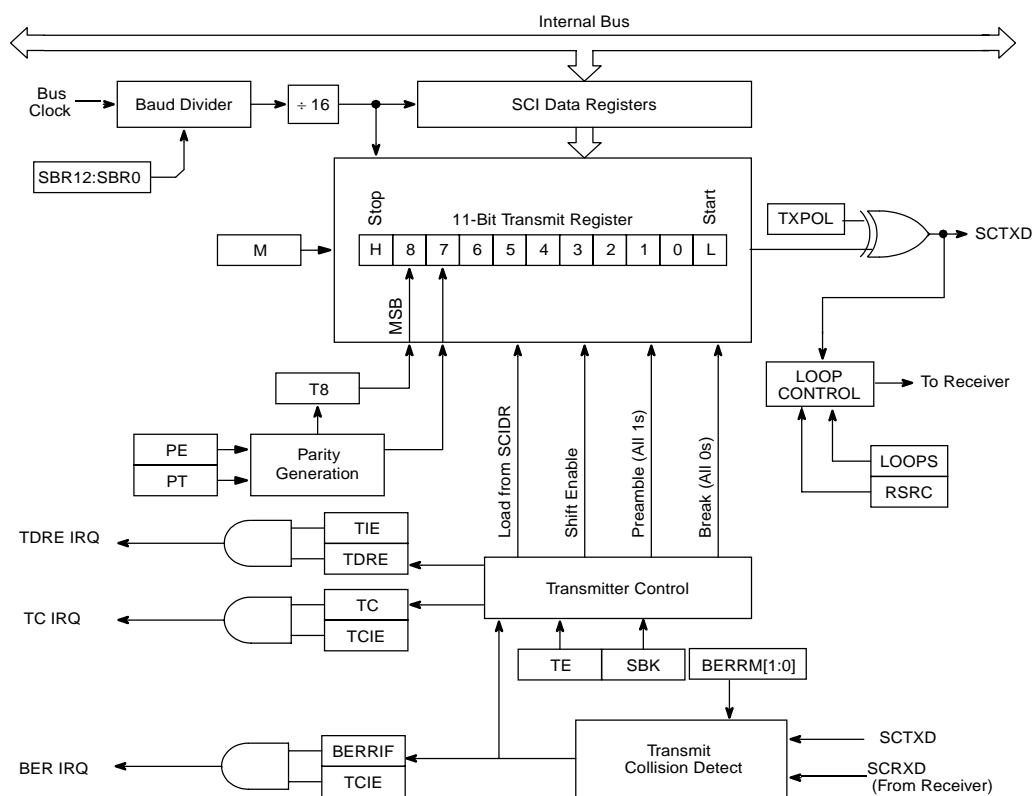


Fig. 4.14. Structura circuitelor de transmisie ale unui canal al modulului de interfață pentru comunicație serială asincronă.

Registrul de date transmisie (*SCI Data Register*) cuprinde registrul SCIDRL în care se scriu biții de date T7÷T0 ai unui caracter de transmis. Pentru comunicații cu 9 biți de date, registrul de date transmisie cuprinde și bitul SCIDRH.6 în care se scrie bitul T8 al unui caracter de transmis. În condițiile de validare a transmisiei, conținutul

registrului de date transmisie se transferă în registrului de deplasare transmisie (*11-Bit Transmit Register*) care, prin deplasare cu frecvența de comunicație f_c , realizează conversia paralel-serie și comandă ieșirea de transmisie date (*SCTDX*).

Bitul M (*Data Format Mode Bit*), SCICR1.4, la nivel logic 0/1, selectează numărul de biți de date dintr-un caracter 8/9.

Bitul PE (*Parity Enable Bit*), SCICR1.1, la nivel logic 1/0, validează/invalidează generarea bitului de paritate la transmisie și verificarea parității la recepție.

Bitul PT (*Parity Type Bit*), SCICR1.0, la nivel logic 0/1, selectează tipul de paritate pară/impară, referitoare la numărul de biți de date (inclusiv cel de paritate) cu nivel logic 1.

Bitul TXPOL (*Transmit Polarity*), SCISR2.4, la nivel logic 0/1, selectează polaritatea normală/inversată a datelor la pinul TXD. Polaritatea normală corespunde reprezentării datelor din figura 4.12.

Bitul TE (*Transmitter Enable Bit*), SCICR2.3, la nivel logic 1/0, validează/invalidează transmisia.

Bitul TDRE (*Transmit Data Register Empty Flag*), SCISR1.7, se poziționează la nivel logic 1 după încărcarea registrului de deplasare transmisie (*11-Bit Transmit Register*) cu biții de date ai unui cuvânt de transmis din registrul SCIDRL. Rezultă că nivelul logic 1 în bitul TDRE indică golirea registrului de date transmisie (*SCI Data Register*) și posibilitatea de încărcare în acesta a unui nou cuvânt din mesajul de transmis. Anularea bitului TDRE se obține după citirea registrului SCISR1 cu bitul TDRE la nivel logic 1, prin scriere în registrul SCIDRL.

Bitul TIE (*Transmitter Interrupt Enable Bit*), SCICR2.7, la nivel logic 1, validează generarea cererilor de întrerupere (*TDRE IRQ*) la momentele de golire a registrului de date transmisie, momente date de poziționările la nivel logic 1 ale bitului TDRE.

Bitul TC (*Transmit Complete Flag*), SCISR1.6, se poziționează la nivel logic zero pe durata transmisiei unor cuvinte succesive dintr-un mesaj și se poziționează la nivel logic 1 la sfârșitul transmisiei mesajului (bitul TDRE este la nivel logic 1 și ultimul cuvânt din mesaj a fost transmis).

Bitul TCIE (*Transmission Complete Interrupt Enable Bit*), SCICR2.7, la nivel logic 1, validează generarea unei cereri de întrerupere (*TC IRQ*) la momentul de sfârșit al transmisiei unui mesaj, momente dat de poziționarea la nivel logic 1 a bitului TC.

Pentru transmisia unui mesaj printr-un canal SCI este necesară configurarea acestuia privind frecvența de comunicație, numărul biților de date, paritate, întreruperi și validarea transmisiei. Transmisia fiecărui cuvânt din mesaj necesită operațiile:

- testarea în buclă (periodică) a bitului TDRE până la citirea acestuia cu nivel logic 1 sau acceptarea cererii de întrerupere (*TDRE IRQ*);

- scrierea următorului cuvânt de transmis în registrul de date transmisie (*SCI Data Register*).

În cele ce urmează se face o prezentare parțială a structurii și funcționării circuitelor de recepție ale unui canal SCI, figura 4.15.

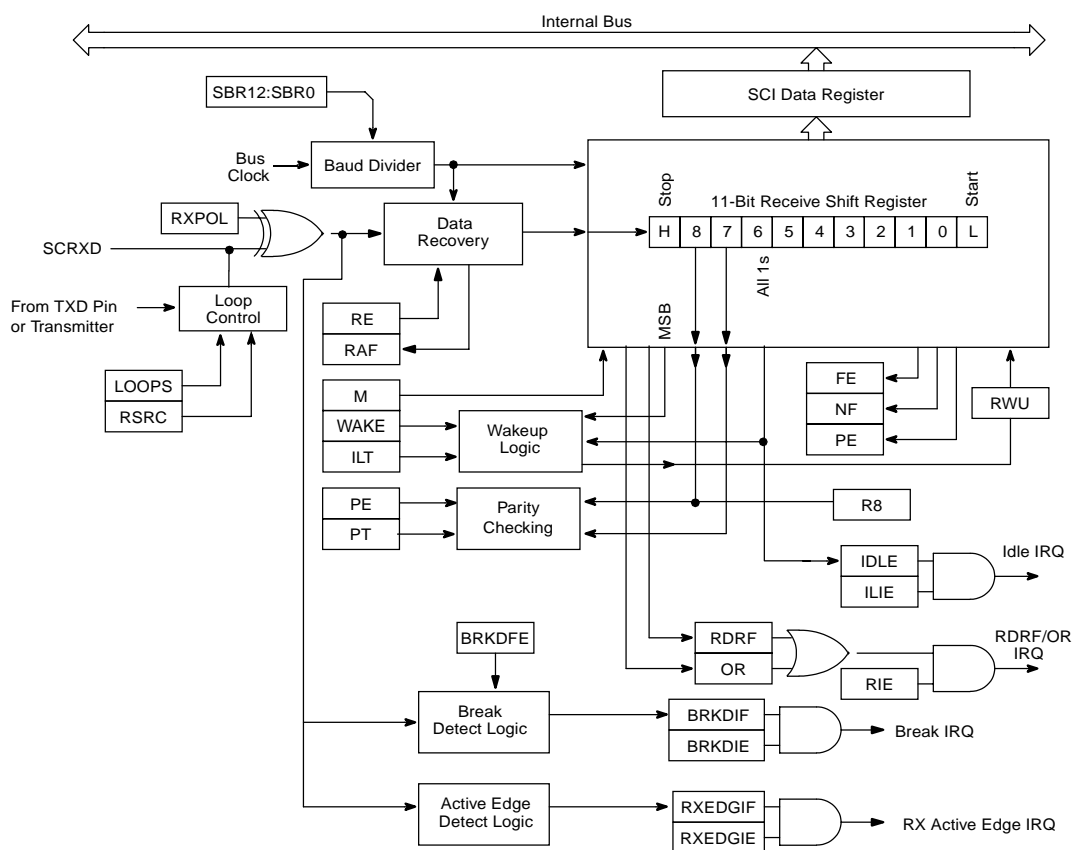


Fig. 4.15. Structura circuitelor de recepție ale unui canal al modului de interfață pentru comunicație serială asincronă.

Pentru recepția serială a unui cuvânt, semnalul de la intrarea de recepție date (*SCRXD*) este eșantionat pentru identificarea bitului de start și, după validarea recepției acestuia, a biților de date și a bitului de stop. Operațiile de eșantionare și identificare se realizează cu circuitul de identificare date (*Data Recovery*). Secvența de biți identificată este transferată în registrul de deplasare recepție (*11-Bit Receive Shift Register*) și, apoi, biții de date sunt transferați în registrul de date recepție (*SCI Data Register*). Registrul de date recepție cuprinde registrul SCIDRL din care se citește biții de date R7÷R0 ai unui caracter recepționat. Pentru comunicații cu 9 biți de date, registrul de date recepție cuprinde și bitul SCIDRH.7 din care se citește bitul R8 al unui caracter recepționat.

Bitul RXPOL (*Receive Polarity*), SCISR2.3, la nivel logic 0/1, selectează polaritatea normală/inversată a datelor de la pinul RXD. Polaritatea normală corespunde reprezentării datelor din figura 4.12.

Bitul RE (*Receiver Enable Bit*), SCICR2.2, la nivel logic 1/0, validează/invalidează recepția.

Bitul RDRF (*Receive Data Register Full Flag*), SCISR1.5, se poziționează la nivel logic 1 după transferul biților de date ai unui cuvânt recepționat din registrul de deplasare recepție (11-*Bit Receive Shift Register*) în registrul de date recepție (*SCI Data Register*). Rezultă că nivelul logic 1 în bitul RDRF indică prezența unui cuvânt recepționat în registrul de date recepție și posibilitatea de citire a acestuia. Anularea bitului RDRF se obține după citirea registrului SCISR1 cu bitul RDRF la nivel logic 1, prin citire din registrul SCIDRL.

Bitul OR (*Overrun Flag*), SCISR1.3, se poziționează la nivel logic 1 dacă asamblarea unui nou caracter în registrul de deplasare recepție se termină înainte de citirea registrului de date recepție.

Bitul RIE (*Receiver Full Interrupt Enable Bit*), SCICR2.5, la nivel logic 1, validează cererile de întrerupere (RDRF/OR IRQ).

Frecvența de eșantionare a semnalului de la intrarea de recepție date (*SCRXD*) este $16 \cdot f_c$ (*RT Clock*), figura 4.16.

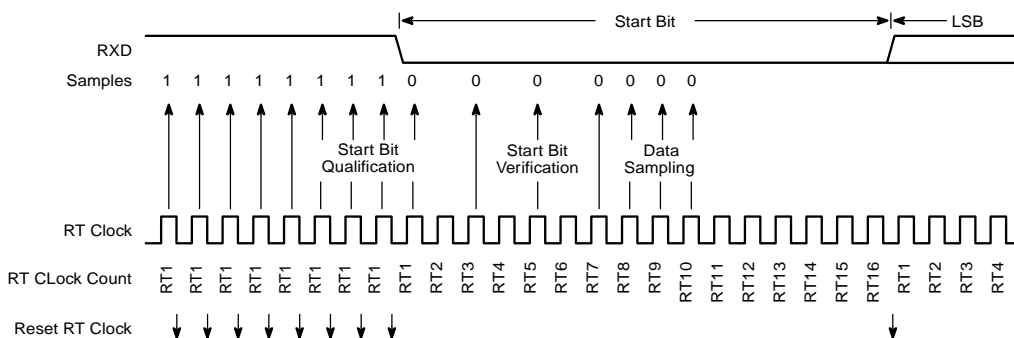


Fig. 4.16. Eșantionarea datelor în recepția serială asincronă.

Detecția frontului de cădere al bitului de start corespunde la două eșantioane succesive cu niveluri logice 1 respectiv 0. Din momentul detecției acestui front, impulsurile cu frecvența de eșantionare sunt contorizate de la RT1 la RT16, figura 4.16, corespunzător unei divizări cu 16 a frecvenței de eșantionare pentru obținerea frecvenței de comunicație f_c . Decizia privind detecția bitului de start se ia pe baza eșantioanelor corespunzătoare impulsurilor RT3, RT5 și RT7 prin logică majoritară. Nivelul logic al unui bit de date și al bitului de stop se stabilește pe baza eșantioanelor corespunzătoare impulsurilor RT8, RT9 și RT10 prin logică majoritară.