

O altă alternativă de identificare este prin *polling software*. La detectarea unei întreruperi, UCP execută o rutină de tratare a întreruperii a cărei sarcină este de a interoga fiecare modul de I/E pentru a determina modulul care a generat întreruperea. Pentru interogare se poate utiliza o linie de comandă separată (de exemplu, TEST I/E). În acest caz, UCP activează această linie și depune adresa unui anumit modul de I/E pe liniile de adresă. Modulul de I/E răspunde dacă a generat întreruperea. O altă posibilitate este ca fiecare modul de I/E să conțină un registru de stare adresabil. UCP citește registrul de stare al fiecărui modul de I/E pentru a identifica modulul care a generat întreruperea.

Dezavantajul metodei de *polling software* este că se consumă un anumit timp al UCP. O tehnică mai eficientă este utilizarea unui lanț de dispozitive (*daisy chain*), care permite de fapt un *polling hardware*. Toate modulele de I/E partajează o linie comună de cerere de întrerupere.

Linia de achitare a întreruperii este înlănțuită prin toate modulele. La detectarea unei cereri de întrerupere, UCP activează un semnal de achitare a întreruperii. Acest semnal se propagă printr-o serie de module de I/E până când ajunge la modulul solicitant, care răspunde de obicei prin plasarea unui vector de întrerupere pe magistrala de date, reprezentând adresa modulului de I/E sau un alt identificator unic. UCP utilizează vectorul ca un pointer la rutina de tratare corespunzătoare modulului respectiv. Astfel se elimină necesitatea execuției prealabile a unei rutine generale de tratare a întreruperii.

Arbitrajul de magistrală utilizează de asemenea întreruperile vectorizate. Cu această tehnică, un modul de I/E trebuie să preia mai întâi controlul asupra magistralei pentru a putea activa linia de cerere a întreruperii. Deci, un singur modul poate activa linia la un moment dat. La detectarea întreruperii, UCP activează linia de achitare, iar modulul care a efectuat cererea își plasează vectorul pe liniile de date.

În cazul cererilor simultane, este necesară stabilirea unor priorități, și determinarea condițiilor în care UCP poate fi întrerupt în timpul servirii unei întreruperi. Perifericele care permit transferuri rapide primesc o prioritate înaltă, iar perifericelor lente li se atribuie o prioritate mai redusă.

2.2.3. Sisteme de întreruperi cu priorități

Stabilirea priorității întreruperilor simultane se poate realiza prin *software* sau prin *hardware*.

În cazul metodei *software*, identificarea sursei de prioritate maximă se realizează prin *polling*. Pentru toate întreruperile, există o rutină comună de servire, care interoghează sursele de întrerupere. Ordinea în care sunt interogate sursele determină prioritatea lor. Dacă cererea de întrerupere a unei surse este activată, se execută saltul la rutina de tratare corespunzătoare. Dezavantajul

metodei software este că în cazul unui număr mare de surse de întrerupere, timpul necesar interogării lor crește.

În cazul metodei *hardware*, se utilizează circuite care acceptă cereri de întrerupere de la mai multe surse, determină cererea cu prioritatea maximă, pe baza căreia generează o cerere de întrerupere către procesor. Pentru creșterea vitezei, fiecare sursă de întrerupere are un vector propriu de întrerupere pentru accesul direct la propria rutină de tratare. Pentru această metodă, liniile de întrerupere pot fi conectate fie în paralel, fie în serie, sub forma unui lanț de dispozitive.

2.2.3.1. Conectarea în paralel a liniilor de întrerupere

Această metodă utilizează un *registru de întrerupere* (RINT) ai cărui biți sunt setați separat de cererile de întrerupere ale fiecărui dispozitiv. Prioritatea este stabilită în funcție de poziția biților din registru. În plus poate exista un *registru al măștilor de întrerupere* (RM) prin care se poate controla starea fiecărei cereri de întrerupere. Prin utilizarea registrului mască se pot dezactiva întreruperile mai puțin prioritare în timpul servirii unui dispozitiv mai prioritar. De asemenea, un dispozitiv mai prioritar va putea întrerupe UCP în timpul servirii unui dispozitiv mai puțin prioritar.

Logica de prioritate pentru un sistem cu patru surse de întrerupere este prezentată în Figura 2.4 [11]. Ea constă dintr-un registru de întrerupere ai cărui biți sunt setați de sursele de întrerupere și pot fi resetați prin program. Dispozitivul D_0 este cel mai prioritar. Registrul mască are același număr de poziții ca și registrul de întrerupere. Este posibilă setarea sau resetarea prin program a oricărui bit din registrul mască. Fiecare bit din RINT și bitul corespunzător din RM sunt conectați la o poartă ȘI pentru a genera intrările pentru codificatorul prioritar. Astfel, o întrerupere este recunoscută numai dacă bitul corespunzător din registrul mască este setat la 1 prin program.

Codificatorul prioritar generează doi biți ai vectorului de întrerupere VINT, care este transferat la UCP prin buffere cu trei stări. Validarea bufferelor se realizează prin semnalul de recunoaștere a întreruperii (*INTACK*) de la UCP și prin două bistabile IST și IEN. Bistabilul de *stare a întreruperilor* IST este setat dacă a apărut o întrerupere care nu este mascată. Bistabilul de *validare a întreruperilor* IEN poate fi setat sau resetat prin program pentru a se permite un control global al sistemului de întreruperi.

Codificatorul prioritar implementează funcția de prioritate. Dacă două sau mai multe intrări sunt activate simultan, va fi luată în considerare intrarea cea mai prioritară. Tabelul de adevăr al unui codificator prioritar cu patru intrări este prezentat în Tabelul 2.1.

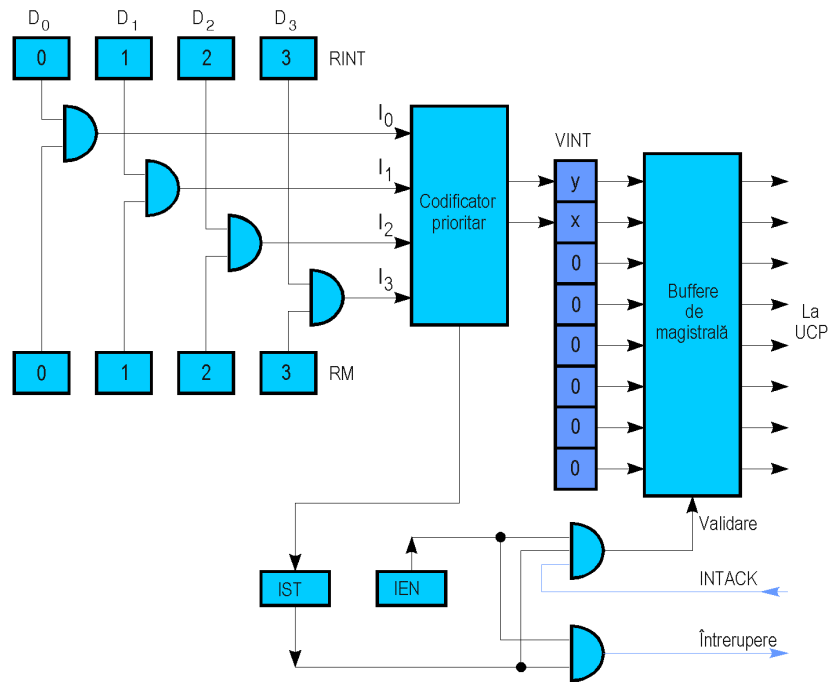


Figura 2.4. Conectarea în paralel a liniilor de întrerupere.

Intrarea I_0 este cea mai prioritară, deci dacă această intrare este 1, se va genera la ieșire $xy = 00$, indiferent de valorile celorlalte intrări. Intrarea I_1 are următorul nivel de prioritate. Ieșirea este $xy = 01$ dacă $I_1 = 1$, cu condiția ca $I_0 = 0$, indiferent de valorile celorlalte două intrări mai puțin prioritare. Bistabilul IST este setat numai dacă una sau mai multe intrări au valoarea 1. Dacă toate intrările sunt 0, IST este șters și celelalte ieșiri ale codificatorului nu sunt utilizate. Aceasta deoarece vectorul de întrerupere nu este transferat la UCP atunci când $IST = 0$.

Ieșirea codificatorului prioritar se utilizează pentru a forma o parte a vectorului de întrerupere pentru fiecare sursă. Celorlalți biți ai vectorului li se poate asigura orice valoare. De exemplu, vectorul de întrerupere se poate forma prin adăugarea unor zerouri la ieșirile x și y ale codificatorului. În acest caz, vectorii de întrerupere ai dispozitivelor de I/E din figură vor avea asigurate valorile 0, 1, 2 și 3.

Tabelul 2.1. Tabelul de adevăr al codificatorului prioritar.

Intrări				Ieșiri		
I_0	I_1	I_2	I_3	x	y	IST
1	X	X	X	0	0	1
0	1	X	X	0	1	1
0	0	1	X	1	0	1
0	0	0	1	1	1	1
0	0	0	0	X	X	0

2.2.3.2. Conectarea în serie a liniilor de întrerupere

Această metodă de selecție a priorității constă în conectarea în serie a tuturor dispozitivelor care pot genera o cerere de întrerupere. Dispozitivul cu prioritatea maximă este plasat în prima poziție, ultimul din lanț fiind dispozitivul cu prioritatea minimă. Această metodă de conectare a trei dispozitive este prezentată în Figura 2.5.

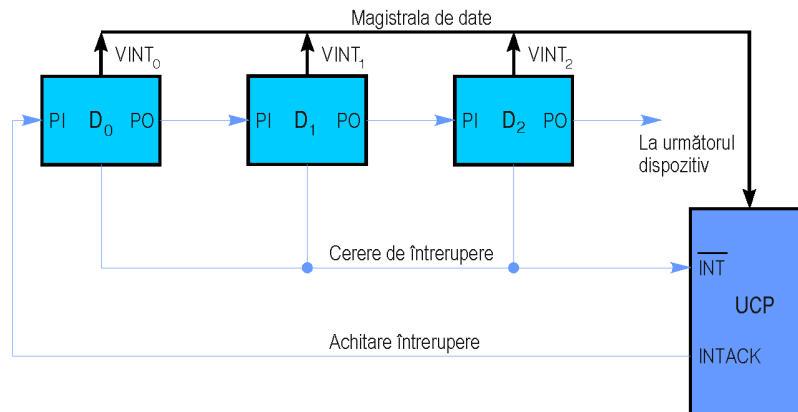


Figura 2.5. Conectarea în serie a surselor de întrerupere.

Linia de cerere a întreruperii este comună pentru toate dispozitivele, formând o conexiune SAU cablat. Dacă nu există nici o cerere de întrerupere, linia de întrerupere rămâne în starea 1 logic. Dacă cel puțin un dispozitiv solicită o întrerupere, linia de cerere trece în starea 0 logic, și validează intrarea de întrerupere a UCP. UCP răspunde la o cerere de întrerupere prin activarea semnalului de achitare *INTACK*. Acest semnal este recepționat de dispozitivul D_0 la

intrarea sa PI (*Priority In*). Semnalul de achitare este transmis mai departe la următorul dispozitiv prin conexiunea PO (*Priority Out*) numai dacă D_0 nu solicită o întrerupere. Dacă acest dispozitiv are o cerere de întrerupere activată, blochează transmiterea semnalului de achitare la următorul dispozitiv prin plasaarea valorii 0 logic la ieșirea sa PO . Apoi D_0 va depune vectorul de întrerupere ($VINT$) pe magistrala de date pentru a fi utilizat de UCP în timpul ciclului de întrerupere.

Un dispozitiv căruia i se aplică un semnal 0 la intrarea sa PI generează un semnal 0 la ieșirea sa PO pentru a informa următorul dispozitiv asupra faptului că semnalul de achitare a fost blocat. Un dispozitiv cu un semnal 1 la intrarea sa PI care solicită o întrerupere, va intercepta semnalul de achitare prin plasaarea unui semnal 0 la ieșirea sa PO . Dacă dispozitivul nu solicită o întrerupere, va transmite semnalul de achitare la următorul dispozitiv prin generarea unui semnal 1 la ieșirea PO . Astfel, dispozitivul cu $PI = 1$ și $PO = 0$ este cel care are prioritatea maximă dintre cele care solicită o întrerupere, și acest dispozitiv va plasa vectorul său de întrerupere pe magistrala de date.

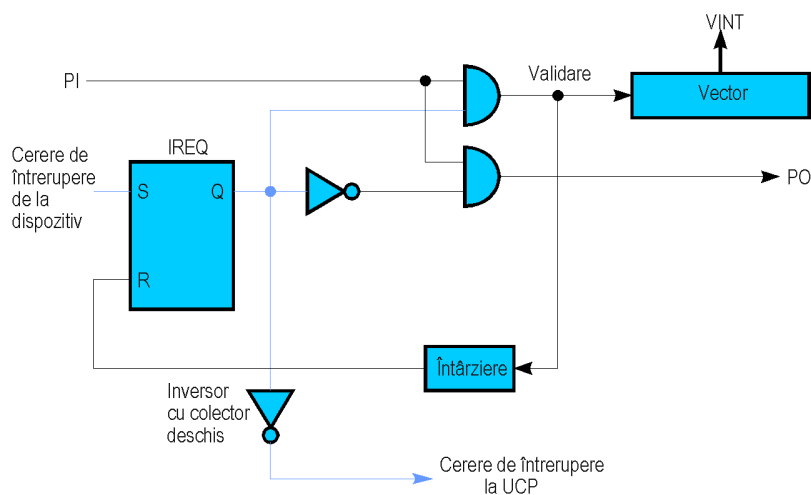


Figura 2.6. Logica internă a unui etaj al sistemului de priorități *daisy chain*.

Figura 2.6 indică logica internă care trebuie conținută de fiecare dispozitiv conectat într-un sistem de priorități *daisy chain*. Dispozitivul setează bistabilul său IREQ atunci când solicită o întrerupere de la UCP. Ieșirea acestui bistabil trece printr-un inversor cu colector deschis pentru a asigura conexiunea SAU cablată necesară pentru linia comună de întrerupere. Dacă $PI = 0$, atât PO cât și semnalul de validare a vectorului $VINT$ sunt 0, indiferent de starea bista-

bilului IREQ. Dacă $PI = 1$ și $IREQ = 0$, rezultă că $PO = 1$ și vectorul este invalidat. Astfel se transmite semnalul de achitare la următorul dispozitiv prin ieșirea PO . Dispozitivul este activ atunci când $PI = 1$ și $IREQ = 1$. Această condiție plasează un semnal 0 la ieșirea PO și validează vectorul de întrerupere. Se presupune că fiecare dispozitiv are propriul său vector de întrerupere. Bistabilul IREQ este resetat după o întârziere suficientă pentru a asigura recepția vectorului de către UCP.

2.2.4. Rutine de tratare a întreruperilor

La sfârșitul fiecărui ciclu de execuție a instrucțiunilor, UCP verifică dacă întreruperile sunt validate (bistabilul IEN este setat) și testează dacă există o cerere de întrerupere. Dacă $IEN = 1$ și există o cerere activă, UCP va executa un ciclu de întrerupere. În timpul acestui ciclu, se execută următoarele operații:

1. Se salvează contorul de program (PC) în stivă.
2. Se activează semnalul de achitare a întreruperii *INTACK*.
3. Se transferă adresa rutinei de tratare în PC, pe baza vectorului de întrerupere.
4. Se dezactivează sistemul de întreruperi, prin resetarea bistabilului IEN.
5. Se trece la ciclul de extragere a primei instrucțiuni din rutina de tratare.

În cadrul fiecărei rutine de tratare a întreruperii trebuie să se execute un set de operații pentru controlul registrelor sistemului de întreruperi. Pentru a se valida cererile de prioritate mai mare decât cea a întreruperii în curs, bistabilul IEN trebuie setat din nou, după ce întreruperile mai puțin prioritare vor fi dezactivate. Principalele operații care trebuie executate la începutul unei rutine de tratare sunt următoarele:

1. Ștergerea biților din registrul mască corespunzători nivelelor mai puțin prioritare.
2. Ștergerea bistabilului IST care indică starea întreruperilor.
3. Salvarea registrelor UCP.
4. Validarea întreruperilor prin setarea bistabilului IEN.

La sfârșitul fiecărei rutine de tratare a unei întreruperi trebuie să se prevadă instrucțiuni pentru a executa următoarele operații:

1. Dezactivarea întreruperilor, prin resetarea bistabilului IEN.
2. Refacerea conținutului registrelor UCP.
3. Ștergerea bitului din registrul cererilor de întrerupere corespunzător sursei de întrerupere deservite.