TECNICHE DI INTERRUZIONE NEI MICROCOMPUTER

Abbiamo analizzato nel modulo E la procedura con cui avviene lo scambio di informazioni in un microcomputer tra la CPU e la memoria. Il μP, che svolge la funzione di master, impone i tempi e i modi con i quali eseguire il trasferimento dei dati dal μP alla memoria e dalla memoria al μP. La CPU genera tutti i segnali di controllo necessari affinché la memoria, sia se deve ricevere sia se deve trasmettere, soddisfi la richiesta del μP. Questo scambio non richiede infatti particolari procedure poiché la memoria, attivato l'ingresso di CE (Chip Enable) o CS (Chip Select), è sempre disponibile al dialogo. Tutto al più potrebbe non essere tanto veloce nel rendere disponibili i dati all'uscita, in tal caso il μP introduce nel ciclo macchina dei tempi di attesa.
Lo scambio di dati con gli organi periferici (tastiera, stampante, ecc.) può invece presentare alcune situazioni particolari per cui il dispositivo d'ingresso o di uscita può non essere disponibile al dialogo. In tal caso la CPU deve verificare, leggendo lo stato di un apposito registro del dispositivo, la disponibilità al dialogo.
Le tecniche di comunicazioni tra il μP e i dispositivi di I/O sono fondamentalmente due: il polling e l'interrupt.

POLLING E INTERRUPT

In un μC lo scambio di dati tra la CPU e la memoria non esige particolari procedure, infatti se in un qualsiasi istante la CPU attiva gli ingressi di abilitazione, la memoria si dichiara immediatamente disponibile: lo scambio dei dati può avvenire.
Lo scambio di informazioni con i dispositivi periferici non è possibile in un qualsiasi momento, ma solo quando gli organi periferici segnalano la loro disponibilità. Ad esempio un processo industriale potrebbe in un certo istante non avere la necessità (o la possibilità) di trasmettere (o ricevere) dati alla (o dalla) CPU; un altro esempio potrebbe essere costituito da una stampante che non può accettare altri dati dalla CPU perché la sua memoria è satura. Pertanto prima di dar luogo alla procedura di scambio dei dati, il μP deve accertare la disponibilità della periferica alla comunicazione.
Esaminiamo le due tecniche di comunicazione tra la CPU e una periferica: il polling e l'interrupt.

Il polling consiste nella scansione ciclica, da parte della CPU, di tutte le periferiche per verificare la disponibilità o meno alla comunicazione.

La tecnica del polling ha l'indubbio vantaggio di una semplicità hardware, infatti ogni periferica è dotata di uno o più registri di stato su cui annota la situazione in cui si trova: libera, occupata, disponibile a inviare un dato o disponibile ad accettare un dato. Pertanto la CPU può consultare il registro di stato della periferica come una qualsiasi locazione di memoria, e conseguentemente assumere le decisioni opportune.
Gli inconvenienti di questo tipo di tecnica possono essere così riassunti:
1) la CPU è impegnata per la quasi totalità del suo tempo in operazioni di verifica dei

registri di stato dei dispositivi di I/O; la comunicazione infatti occupa un tempo notevolmente inferiore rispetto alla interrogazione sequenziale di tutti i dispositivi;
2) un richiesta di dialogo con carattere di urgenza, avanzata da una periferica alla CPU, non può essere soddisfatta immediatamente perché il μP è impegnato nella scansione delle altre periferiche; la comunicazione potrebbe avvenire con un ritardo intollerabile se si tratta di un processo industriale da controllare;
3) l'intervallo di tempo che intercorre tra due verifiche successive della stessa periferica non sarà mai fisso perché tale tempo dipende dallo stato delle altre periferiche e quindi dal tempo impiegato dal μP per soddisfare la richiesta di dialogo della periferica disponibile. In alcuni casi questo intervallo di tempo può essere troppo lungo e quindi non accettabile. Si pensi ad esempio a un sistema dotato di trasmissione seriale sincrona: se il dato non giunge in tempo utile, la periferica riceve un segnale errato.

La tecnica dell'interrupt (interruzione) consiste nella richiesta di dialogo avanzata dalla periferica alla CPU mentre questa è impegnata nello svolgimento di altri compiti.

Il μP è infatti provvisto di un ingresso INTRQ (Interrupt Request = Richiesta d'Interruzione) che viene attivato dal registro di stato della periferica. La CPU, alla fine dell'esecuzione dell'istruzione in corso di elaborazione e prima della fase di fetch dell'istruzione successiva, risponde al dispositivo, che ha avanzato la richiesta, con un segnale di INTA (Interrupt Acknowledge = Riconoscimento dell'Interruzione) e svolge la routine di servizio dell'interrupt, cioè esegue un segmento di programma, già memorizzato in precedenza nel sistema, atto a soddisfare l'esigenza del dispositivo richiedente l'interruzione. Dopo aver svolto la routine di servizio, il μP torna al programma precedente esattamente al punto in cui aveva sospeso l'esecuzione.
È necessario sottolineare che il funzionamento di una periferica è asincrono rispetto a quello della CPU pertanto la richiesta di interruzione può giungere all'unità centrale in qualsiasi istante.
Con questa tecnica la CPU dedica alle periferiche il tempo strettamente necessario allo scambio dei dati, mentre la sua attività principale è rivolta allo svolgimento di altre mansioni.
Poiché la richiesta di interruzione viene presa in considerazione solo alla fine dell'esecuzione dell'istruzione attualmente in corso, si rende necessario memorizzare tale richiesta fino all'istante in cui il μP è in grado di soddisfarla. Pertanto il microprocessore deve essere dotato di un elemento di memoria per immagazzinare il segnale attivo di richiesta di interruzione.
Se al microprocessore provengono dalle periferiche più richieste di interruzione sorgono due problemi:

1. le molteplici richieste devono giungere sull'unico ingresso della CPU;
2. è necessario individuare la periferica che ha richiesto l'interruzione.

Per quanto riguarda il primo punto, una delle possibili soluzioni è indicata in figura 1 avendo ipotizzato che i segnali INT delle periferiche siano attivi a livello basso. Se invece tali segnali sono attivi a livello alto la porta AND deve essere sostituita con una NOR.

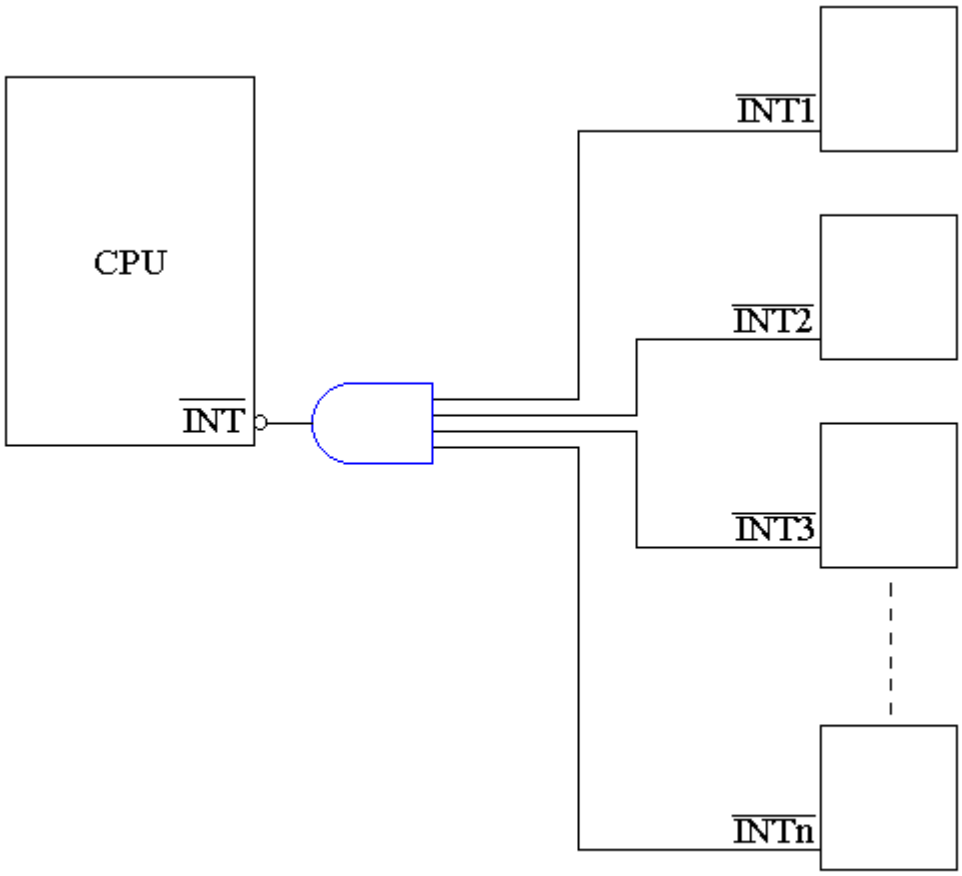

Figura 1

Per quanto riguarda l'individuazione della periferica, il programma di servizio viene caricato con un indirizzo dedotto da informazioni fornite dallo stesso dispositivo che ha richiesto l'interruzione. Il microprocessore risponde alla richiesta con un segnale di riconoscimento dell'interrupt (Interrupt Acknowledge). In figura 2 è riportato un possibile schema di collegamento tra una CPU e le periferiche.
In realtà la struttura è più complessa di quanto appare in figura perché il blocco di controllo e gestione delle interruzioni deve rispettare i legami temporali impostati dal microprocessore. Questa struttura di collegamento con la CPU è di solito interna ai dispositivi di I/O che fungono da interfaccia tra le periferiche e la CPU.

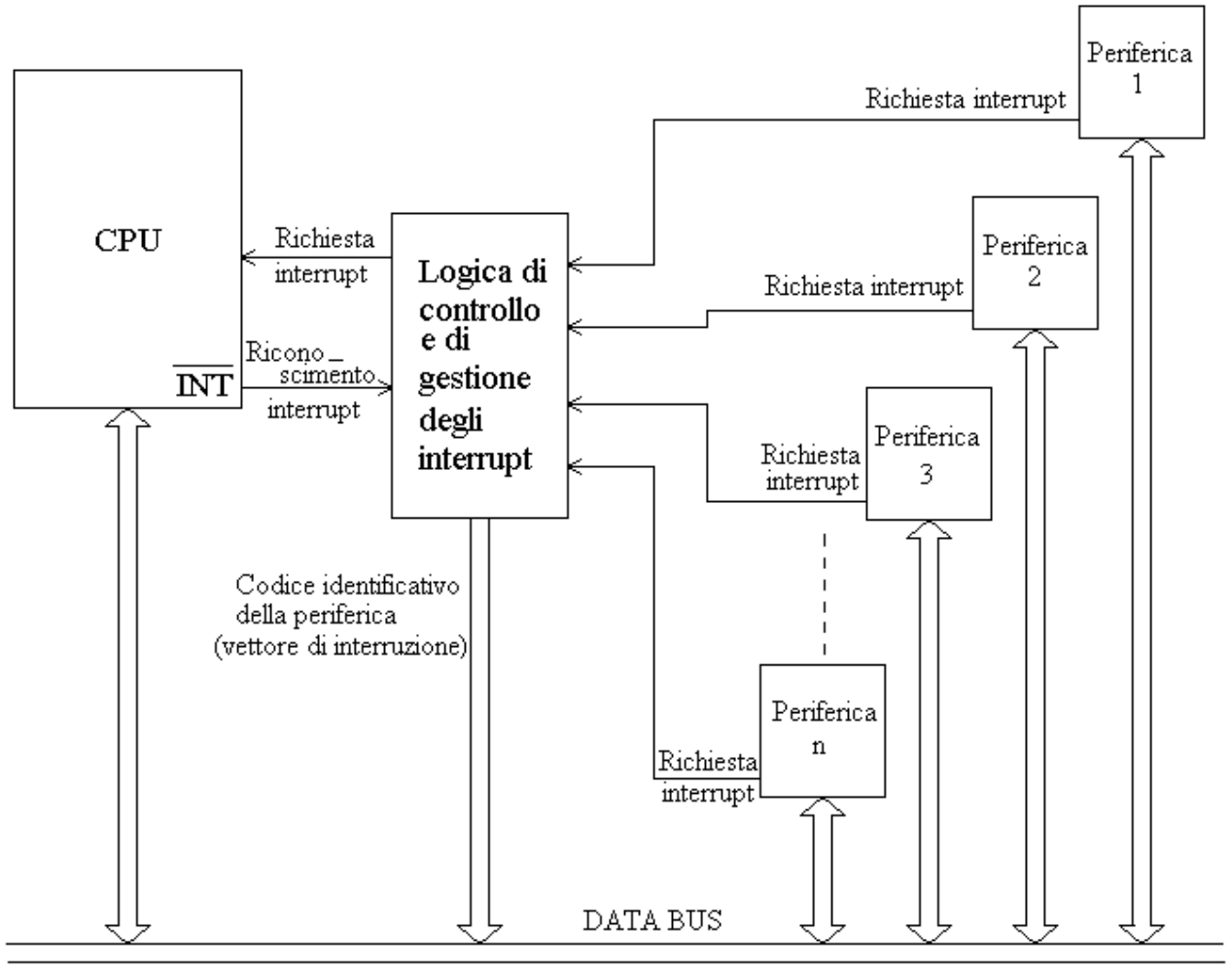

Figura 2

Richieste multiple di interrupt

In un sistema in cui sono presenti più dispositivi di ingresso e di uscita in grado di richiedere un colloquio con la CPU, è necessario individuare quale periferica deve essere soddisfatta e quindi attivare il relativo sottoprogramma di servizio.

Se il μP ha tanti ingressi di interrupt quante sono le periferiche, l'individuazione della periferica che ha richiesto il colloquio è immediato, e tra l'altro non richiede un hardware aggiuntivo.

Quando più periferiche sono collegate allo stesso ingresso di interrupt, si pone il problema di individuare quale tra queste ha fatto richiesta di interruzione. In questo caso le tecniche che si possono adottare sono due: il polling e la vettorizzazione degli interrupt.

Quando si utilizza la tecnica del polling (figura 3) ogni dispositivo ha la linea di richiesta collegata tramite una porta logica, che può essere una AND se la linea è attiva a livello baso o una NOR se a livello alto all'ingresso di interrupt della CPU. Il microprocessore inizia la routine di servizio dell'interruzione e successivamente acquisisce dalla porta d'ingresso il codice identificativo della periferica stessa e quindi esegue il sottoprogramma relativo a quella periferica.

In questo tipo di polling non vi è perdita di efficienza del sistema poiché c'è sicuramente una periferica attiva che ha richiesto il dialogo per uno scambio di dati.

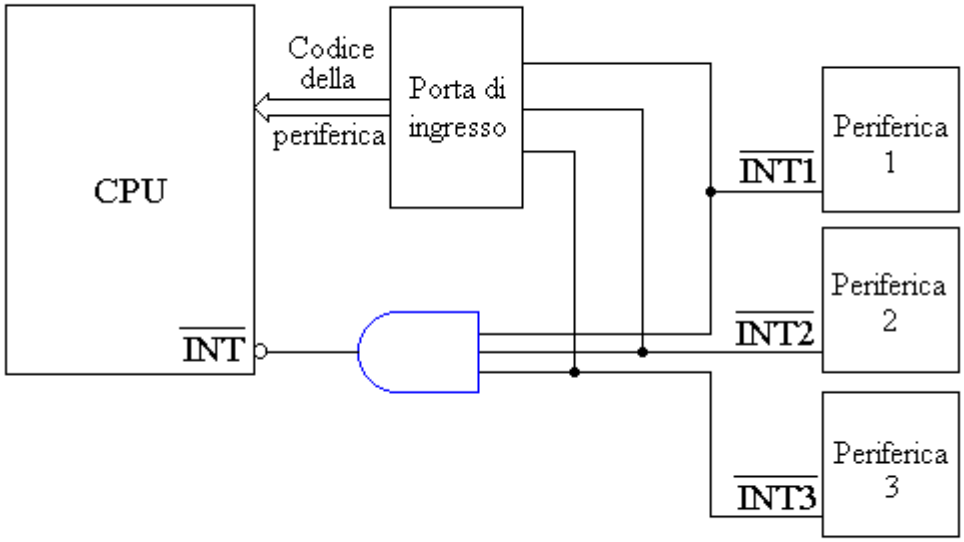

Figura 3

Nella tecnica degli interrupt vettorizzati invece è la stessa periferica che genera direttamente alla CPU un vettore (dato) che serve per formare l'indirizzo di memoria dove ha inizio il sottoprogramma di servizio per la gestione dell'interrupt. In figura 4 è riportato uno schema che illustra questo tipo di tecnica.

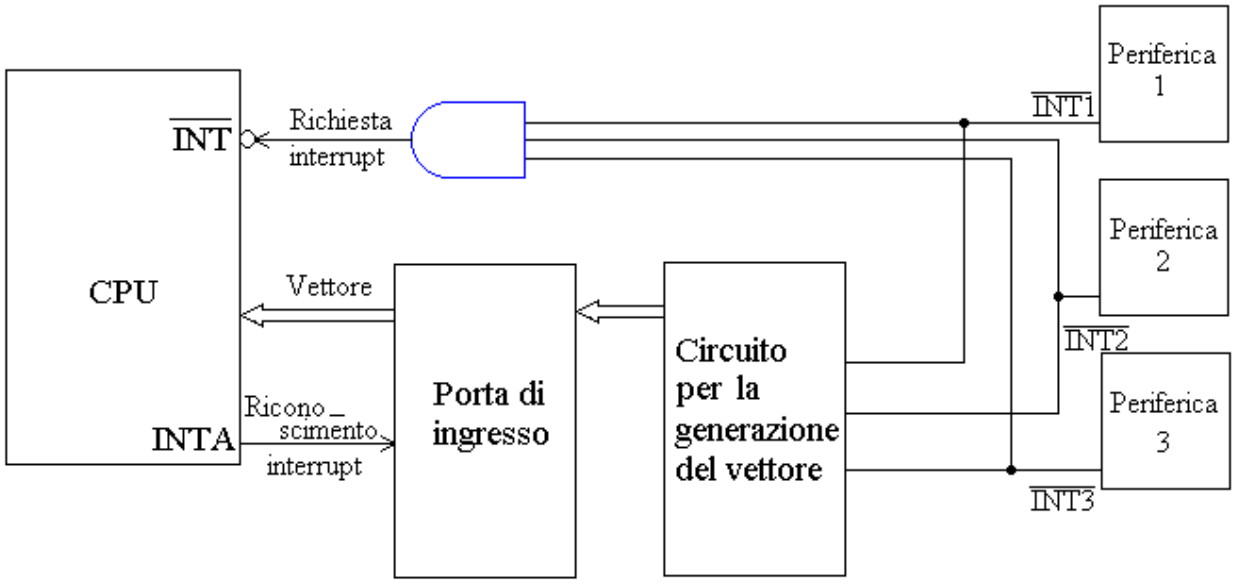

Figura 4

Quando una periferica attiva l'ingresso INT della CPU, viene posto sul Bus Dati, tramite una porta d'ingresso, un vettore di indirizzo. Il µP, dopo aver inviato un segnale di accettazione dell'interrupt, acquisisce il vettore fornito dalla porta

d’ingresso, e forma l’indirizzo di memoria dove si trova la prima istruzione del sottoprogramma di servizio di quella periferica che ha richiesto il dialogo.
Di solito il vettore di interruzione rappresenta una parte o la totalità dell’indirizzo di partenza della subroutine di servizio, ciò dipende dal tipo di microprocessore.
Questa tecnica consente una risposta più rapida da parte della CPU ma richiede un hardware aggiuntivo, quindi più complesso. Infatti il circuito per la generazione del vettore non è semplice da realizzare anche se la soluzione preferita è quella di utilizzare integrati dedicati a questo scopo che tra l‘altro sono anche programmabili, ovvero hanno la possibilità di poter generare il vettore grazie a istruzioni di comando inviati dalla CPU.

Gestione delle priorità
Uno dei problemi che è necessario esaminare è la corretta e efficiente gestione delle richieste di interruzione che arrivano contemporaneamente alla CPU. È utile precisare che le richieste di interrupt si considerano contemporanee se giungono al microprocessore durante l’intervallo di tempo di esecuzione di una istruzione. Infatti quando termina l’esecuzione dell’istruzione, la CPU analizza eventuali richieste e quindi trova attive più periferiche.
Il problema viene risolto assegnando ad ogni sorgente di interrupt un livello di priorità: se le richieste giungono contemporaneamente viene servita quella a priorità più elevata e se il P sta eseguendo la routine di servizio di una interruzione può essere interrotto solo da un interrupt di livello superiore.
La realizzazione pratica di questa strategia può essere differente poiché sono diversi i tipi di microprocessori esistenti in commercio e possono essere differenti le situazioni in cui si deve operare. Tutti i microprocessori sono dotati di istruzioni che consentono di abilitare e disabilitare la loro capacità a rispondere alle richieste di interrupt.
Se il microprocessore è dotato di un solo ingresso di interrupt e può quindi ricevere una sola richiesta alla volta, con un opportuno hardware esterno è possibile gestire e variare la priorità delle richieste. Lo schema di figura 5 indica una possibile soluzione di questo problema. La CPU può mascherare i segnali di INT provenienti dalle periferiche ponendo tramite la porta di uscita a 0 i segnali di gating; viceversa ponendo a 1 gli ingressi G è in grado di attivare singolarmente le richieste. Se il microprocessore sta servendo una richiesta di una periferica con priorità elevata può ignorare, via software, le richieste delle periferiche con priorità più basse. Completata questa operazione la CPU può procedere a riabilitare le richieste con priorità inferiore e servirle se ce n’è una pendente.
Lo schema di figura 5 può essere utile anche per un cambiamento della gestione delle priorità, infatti se il sottoprogramma di servizio di un dispositivo di bassa priorità viene continuamente interrotto da una periferica più elevata, è possibile cambiare l’ordine di servizio poiché i tempi di attesa della periferica bassa possono essere eccessivamente lunghi.
Per soddisfare l’esigenza di servire le periferiche di bassa priorità in tempi accettabili si può cambiare la gestione delle abilitazioni ma ciò richiede una opportuna

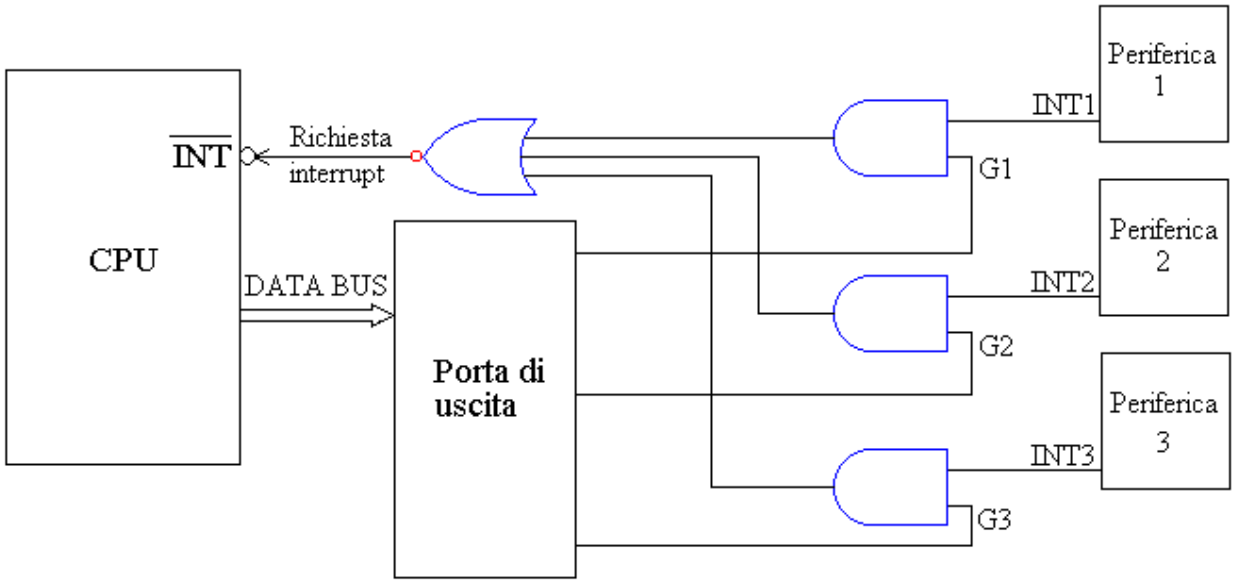

Figura 5

organizzazione del software. Ciò rappresenta un indubbio vantaggio poiché il sistema risulta più flessibile ma può presentarsi l'inconveniente dovuto di un tempo troppo lungo per riprogrammare la porta di uscita prima di mandare in esecuzione il sottoprogramma dell'interrupt pendente.

Tecnica del DAISY CHAIN

L'organizzazione degli interrupt adottata nello schema di figura 5 è chiamata a polling. I livelli di priorità sono definiti via software dai principi adottati per la scansione dei registri di stato delle periferiche, ne consegue una gestione elastica del sistema.

Un altro tipo di organizzazione, realizzata però via hardware, è la tecnica del daisy chain, riportata in figura 6. ogni periferica è dotata di tre linee per la gestione dell'interrupt: INT (richiesta di interrupt), IEI (Interrupt Enable Input = ingresso di abilitazione dell'interruzione), IEO (Interrupt Enable Output = uscita di abilitazione dell'interruzione). Il dispositivo è abilitato alla richiesta se l'ingresso IEI è a livello alto come indicato in figura 6 per la periferica 1, che quindi ha priorità più elevata. Quando IEI è a livello alto, l'uscita IEO viene posta a livello basso inibendo la periferica a priorità immediatamente successiva ad attivare la linea INT. Infatti se IEI è a livello basso anche IEO è a livello basso e il dispositivo non può attivare la INT e inibisce anche la periferica successiva a fare altrettanto.

Una periferica che ha attivato la sua linea INT impedisce a tutti gli altri dispositivi inferiori della catena a richiedere un interrupt. Si stabilisce così una gerarchia nella priorità delle richieste.

Come tutte le soluzioni hardware, la tecnica del daisy chain non consente variazioni nella gestione delle priorità.

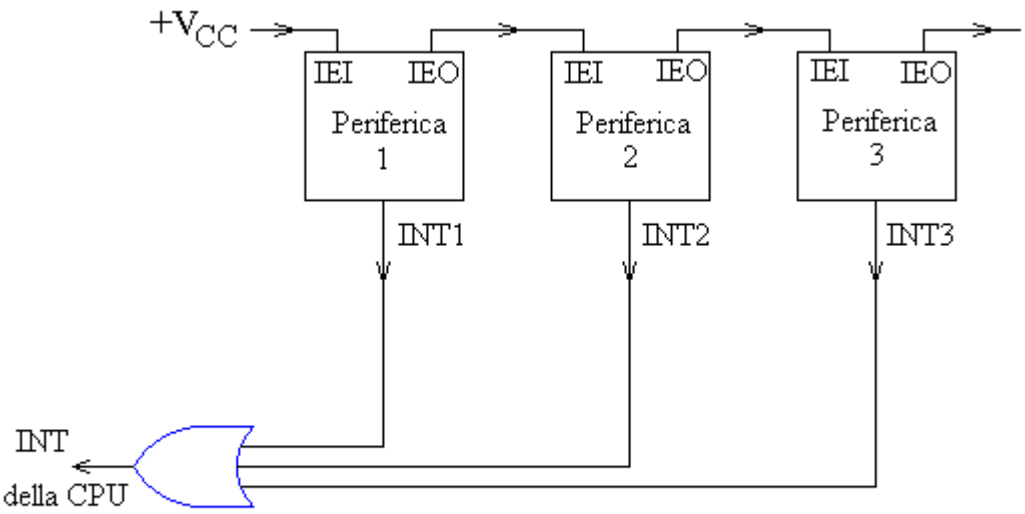

Figura 6

Svantaggi della tecnica degli interrupt

Il colloquio con le periferiche mediante la tecnica degli interrupt è indubbiamente vantaggiosa rispetto al polling ma presenta anche degli svantaggi.

Abbiamo visto che la tecnica dell'interrupt necessita di una struttura hardware più complessa. Tuttavia questo aspetto è poco rilevante perché in pratica si utilizzano circuiti integrati dedicati a tale scopo e quasi sempre integrati nei dispositivi periferici della stessa famiglia del microprocessore.

L'altro svantaggio è costituito dalla difficoltà di verificare la funzionalità di un software legato a un evento asincrono come lo è l'interrupt. Pertanto non è agevole controllare tutte le situazioni possibili che si possono verificare. È necessario quindi adottare particolare attenzione nella stesura delle subroutine di servizio in modo da evitare errori che tra l'altro sono difficili da prevedere proprio per la natura casuale degli interrupt.