

Dispositivi multipli

- ▶ La presenza di più dispositivi di I/O, ognuno con la propria routine di servizio (ISR), pone alcuni problemi
 - ▶ Identificazione del dispositivo che ha emesso la richiesta
 - ▶ Determinazione dell'indirizzo della ISR appropriata
 - ▶ Possibile necessità di interrompere l'esecuzione di una ISR
 - ▶ Possibile simultaneità di richieste da dispositivi diversi
- ▶ Soluzioni per trovare la periferica che ha mandato la richiesta di interruzione
 - ▶ Scansione (polling) del bit **IRQ** nei registri di stato delle interfacce di I/O
 - ▶ **Interruzioni vettorzate**: l'interfaccia di I/O, in risposta alla conferma dell'IRQ, manda un codice di pochi bit (4 o 8), che la **identifica**. Tale codice è l'indice nella **tabella dei vettori d'interruzione** dell'indirizzo della sua ISR. La tabella risiede in memoria nell'intervallo di indirizzi più basso. L'indirizzo contenuto in tabella è detto **vettore di interruzione**, il processore lo legge e lo scrive in **PC** e passa a eseguire la routine

15

Annidamento delle interruzioni

- ▶ Per evitare che una periferica richieda di essere servita mentre si sta servendo un'altra periferica, si possono **disabilitare le interruzioni**. Poiché il tempo di servizio è breve, solitamente, il ritardo introdotto è accettabile per la maggior parte delle periferiche
- ▶ Per alcune periferiche potrebbe essere necessario passare all'esecuzione di un'altra routine di servizio durante l'esecuzione di una routine di servizio
- ▶ Si attribuisce ad ogni periferica un **livello di priorità**, il processore passa a servire una richiesta di interruzione solo se la richiesta proviene da una periferica con livello superiore a quello corrente
- ▶ Non appena il processore comincia a servire la richiesta il livello corrente si alza, ed è pari a quello della periferica. Così si disabilitano le richieste da periferiche di livello di priorità inferiore. Quando il servizio termina, il livello e l'esecuzione ritorna a quello che precedentemente era stato sospeso
- ▶ Quando sono permesse interruzioni annidate, ciascuna routine di servizio deve conservare in pila **PC** e **PS**, prima che la routine di servizio abiliti le interruzioni impostando a 1 **IE**
- ▶ Il livello di priorità corrente è codificato in **bit** appositi del registro **PS**

16

Richieste di interruzione simultanee

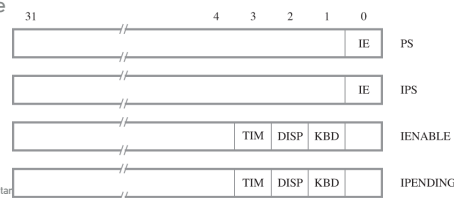
- ▶ Quando due o più richieste arrivano simultaneamente, il processore decide quale richiesta è arrivata per prima tramite uno dei due meccanismi seguenti
 - ▶ Scandendo il registro delle periferiche, quindi l'ordine di scansione indica il livello di priorità della periferica
 - ▶ Usando le interruzioni vettorzate e selezionando una sola periferica per permettere di mandare il vettore di interruzione. I circuiti di arbitraggio (hardware) permettono di selezionare una sola periferica

Controllo della richiesta

- ▶ Il bit di abilitazione di interruzione nel circuito di interfaccia del dispositivo permette di disabilitare o abilitare le richieste di interruzione
- ▶ L'interfaccia della periferica ha un registro di controllo, nel quale vi è il bit **IE** che serve a tale abilitazione
- ▶ Per la tastiera tale bit è **KIE**, e va posto a 1 quando le richieste di interruzioni sono abilitate
- ▶ Come detto, **KIN** è 1 quando un dato è pronto. Quando entrambi **KIN** e **KIE** sono 1, la richiesta viene segnalata ponendo **KIRQ** a 1

Registri di controllo del processore

- ▶ Il registro **PS** include il bit di abilitazione delle interruzioni **IE**
- ▶ Il registro **IPS** è usato per salvare automaticamente i contenuti di **PS** quando una richiesta di interruzione è ricevuta e accettata
- ▶ Se sono permesse interruzione annidate, bisogna salvare in pila il contenuto di **IPS**
- ▶ Il registro **IENABLE** permette di assegnare un bit a ciascun dispositivo, e quando un bit è 1 il processore accetterà le richieste dal dispositivo corrispondente
- ▶ Il registro **IPENDING** indica le richieste di interruzioni attive
- ▶ I registri sono a 32 bit, quindi possono essere gestite fino a 32 periferiche
- ▶ L'accesso a tali registri avviene tramite istruzioni speciali, del tipo



MoveControl R2, PS

- ▶ Carica il contenuto del registro di stato in **R2**

Prof. Tramontana

Routine di servizio

- ▶ Legge il carattere digitato dal registro di interfaccia della tastiera
- ▶ Memorizza il carattere in memoria
- ▶ Visualizza il carattere sullo schermo
- ▶ Quando si riconosce la fine della linea, imposta a 1 la variabile EOL, e disabilita le interruzioni da tastiera
- ▶ Rientra dall'interruzione

Routine di servizio delle interruzioni			
ILOC:	Subtract	SP, SP, #8	Salva i registri
	Store	R2, 4(SP)	
	Store	R3, (SP)	
	Load	R2, PNTR	Carica il puntatore all'indirizzo
	LoadByte	R3, KBD_DATA	Leggi un carattere dalla tastiera
	StoreByte	R3, (R2)	Scrivi il carattere in memoria
	Add	R2, R2, #1	Incrementa il puntatore
	Store	R2, PNTR	Aggiorna il puntatore in memoria
ECO:	LoadByte	R2, DISP_STATUS	Attendi che lo schermo sia pronto
	And	R2, R2, #4	
	Branch_if_[R2]=0	ECO	
	StoreByte	R3, DISP_DATA	Visualizza il carattere appena letto
	Move	R2, #CR	Codice ASCII per il Ritorno Carrello
	Branch_if_[R3]≠[R2]	RTRN	Rientra se non è CR
	Move	R2, #1	
	Store	R2, EOL	Indica la fine della linea
	Clear	R2	Disabilita le interruzioni nell'interfaccia della tastiera
	StoreByte	R2, KBD_CONT	
RTRN:	Load	R3, (SP)	Ripristina i registri
	Load	R2, 4(SP)	
	Add	SP, SP, #8	
	Return-from-interrupt		

Programmi con interruzioni

- ▶ Un programma che legge una linea di caratteri da tastiera e li memorizza in memoria
- ▶ Il programma principale carica l'indirizzo LINEA nella posizione di memoria PTR, usata poi dalla routine di servizio delle interruzioni. Abilita le interruzioni nel registro d'interfaccia della tastiera e nel processore. Abilita il processore ad accettare le interruzioni da tastiera, e le interruzioni in generale

Programma principale (Main)

INIZIO:	Move	R2, #LINEA	
	Store	R2, PNTR	Inizializza il puntatore del buffer
	Clear	R2	
	Store	R2, EOL	Azzerà l'indicatore di fine linea
	Move	R2, #2	Abilita le interruzioni nell'interfaccia della tastiera
	StoreByte	R2, KBD_CONT	
	MoveControl	R2, IENABLE	
	Or	R2, R2, #2	Abilita le interruzioni da tastiera nel registro di controllo del processore
	MoveControl	IENABLE, R2	
	MoveControl	R2, PS	
	Or	R2, R2, #1	
	MoveControl	PS, R2	Poni a 1 il bit di abilitazione delle interruzioni in PS
	Prossima istruzione		

20

Gestore delle interruzioni (interrupt handler)

- ▶ Quando più dispositivi segnalano una richiesta di interruzione, si usa l'informazione nel registro **IPENDING** per scegliere la routine di servizio
- ▶ Il programma Main abilita ciascuna periferica (registri di interfaccia e registro **PS**)
- ▶ Alla richiesta di interruzione, si carica automaticamente l'indirizzo **ILOC** in **PC**
- ▶ Quindi si identifica il dispositivo tramite il registro **IPENDING**

ILOC:	Subtract	SP, SP, #12	Salva i registri
	Store	LINK_reg, 8(SP)	
	Store	R2, 4(SP)	
	Store	R3, (SP)	
	MoveControl	R2, IPENDING	Controlla il contenuto di IPENDING
	And	R3, R2, #4	Controlla se lo schermo ha segnalato la richiesta
	Branch_if_[R3]=0	TESTKBD	Se no, controlla se lo ha fatto la tastiera
	Call	DISR	Chiama la routine di servizio delle interruzioni da schermo (display ISR)
TESTKBD:	And	R3, R2, #2	Controlla se la tastiera ha segnalato la richiesta
	Branch_if_[R3]=0	PROSSIMO	Se no, controlla il prossimo dispositivo
	Call	KISR	Chiama la routine di servizio delle interruzioni da tastiera (keyboard ISR)
PROSSIMO:	...		Controlla le interruzioni da altri dispositivi di I/O
	Load	R3, (SP)	Ripristina i registri
	Load	R2, 4(SP)	
	Load	LINK_reg, 8(SP)	
	Add	SP, SP, #12	
	Return-from-interrupt		

Prof. Tramontana

22

Concetto di eccezione

- ▶ L'interruzione forza la sospensione del programma corrente per iniziarne un altro
- ▶ Il termine **eccezione** è usato per indicare un qualsiasi evento che causa un'interruzione. Quindi le interruzioni da I/O sono un esempio di eccezione
- ▶ Eccezioni da memoria per ripristino da errore: la memoria è dotata di codice per rilevare errore per stabilire se i dati siano alterati, e la presenza di dati alterati è segnalata al processore mandando una richiesta di interruzione
- ▶ Altre eccezioni: tentativo di eseguire un'istruzione con codice operativo inesistente, errore per l'esecuzione di una divisione per zero, tentativo di accesso a parti protette della memoria
- ▶ A seguito di un'eccezione il processore sospende l'esecuzione del programma corrente ed esegue una routine di servizio. Se l'eccezione non proviene da periferica, ovvero è un errore, il processore rinuncia a completare l'esecuzione dell'istruzione corrente
- ▶ Ulteriori eccezioni: esecuzione passo passo (trace) e breakpoint, per programmi di debug; eccezioni da sistema operativo per interagire con i programmi utente