PERIFERICS

1.−INTRODUCCIÓ

1.−INTRODUCCIÓ

Conceptes Basics

El hardware i el nivell de llenguuatge màquina són esencials pel funcionament d'un ordinador. El sistema operatiu permet facilitar la utilització del hardware per part d'un usuari, sense haver de saber les instruccions per fer una operació concreta.

Sempre que tenim un terminal, periferic exterior, hem de tenir una conexió al hardware. Aquesta conexió es podrá gestionar a través de registres.

Els prograames han de ser a memória i hem de poder interactuar amb aquest programa (per executar−lo, escriure−hi alguna dada, etc.), per poder efectuar aquesta interacció amb l'ordinador existeixen uns aparells, que són els anomenats periférics.

El procesador i la memória depenen molt l'un de l'altre i sempre es vol aconseguir fer un programa el més ràpid posible i que ocupi el menys espai possible a la memòria, però en aquests programaes s'han d'introduir dades o treure−les. Aquesta funció d'introduir o extreure dades la porta a terme el subsistema d'entrada/sortida (E/S), i aquest subsistema es qui mana l'entrada de dades i no el procesador. Per tant s'ha d'aconseguir que la comunicació entre E/S i processador sigui la més ràpida possible.

Subsistema d'E/S

Les característiques d'un sistema d'entrada/sortida han de ser les següents:

- Conseguir una baixa interferéncia de les tasques d'E/S amb la resta del sistema.
- Adaptar les senyals entre CPU i periféric. Sempre ens basem en els busos de dades, control i adreces encara que els periférics siguin molt diferents.
- Conversió de la informació a ormats adequats. El format per representar informació a una impresora i a la pantalla és diferent entre ells pot ser diferent a la representació que fa l'ordinador de la mateixa informació.
- Adaptar la velocitat de transferéncia de dades entre CPU i periféric ja que la CPU és molt més ràpida que els perférics.
- El tipus d'informació que intervé en les tasques d'E/S són:

· Dades (caràcter que es vol escriure).

· Control (escriure el caràcter en negreta, el tamany, etc.).

Perifèrics

Un perifèric es pot definir com un dspositiu especialitzat en convertir les dades que li arriben en un llenguatge que enten l'ordinador, o de convertir les dades de l'ordinador en un llenguatge entenible per nosaltres.

Les característiques d'un periféric són les següents:

- Comportament: entrada, sortida o magatzem de dades.
- Usuari: o be és una persona o bé és una màquina.
- Velocitat: freqüència màxima a la que les dades poden ser transferides.

Dispositiu	**Comportament**	**Usuari**	**Velocitat (Kb/seg)**
Teclat	Entrada	Persona	0.01
Ratolí	Entrada	Persona	0.02
Veu	Entrada	Persona	0.2
Impresora laser	Sortida	Persona	100
Pantalla gràfica	Sortida	Persona	30000
Disquette	E/S	Màquina	50
Cinta magnètica	E/S	Màquina	2000
Disc Optic	E/S	Màquina	500
Disc Magnètic	E/S	Màquina	2000

Controladors

Els controladors han de realitzar les següents operacions:

- Adaptar el senyals que surten del subsistema d'E/S al perifèric. Facilitar l'interconexió entre l'ordinador i el perifèric i viceversa.
- Posibilitat de programar−se amb paràmetres.

Estructura del subsistema d'E/S

Els ordinadors es poden dividir en dos grans grups: els PC o ordinadors perssonals, el qual normalment només l'utlitza una persona; i els ordinadors que considerarem grans, els quals són multiusuari.

Els PC s'organitzen amb controladors (el controlador d'interrupcions, el de DMA, i els dels perifèrics que tingui), i juntament amb la CPU i el sistema de memòria, tots tres units pels busos d'adreces, dades i control.

En canvi els ordinadors grans, s'organitzen, enlloc de amb bussos, amb canals, on cada canal tindrá el seu controlador i a partir d'aquest controlador sortirán els controladors de perifèrics, DMA, etc.

Amb els controladors de canals s'en paqueten les dades per poder−les enviar d'un sol cop pel canal, d'aquest manera es vol incrementar la velocitat de comunicació entre els bloc del ordinadors.

Els microcontroladors són els que realment controlen els perifèrics. Aquests es veuen com un PC amb els seus perifèrics, ja que són aquests qui els controlen al igual que el PC, però els microncontroladors estan dins del PC.

Conexió d'un peifèric al sistema

Els controladors es veuen per la CPU com una serie de registres ja siguin d'estat, dades o de control). Nosaltres hem de poder accedir a aquests registres i per poder accedir els registres estàn mapejats, j asigui a memòria principal o a l'espa d'E/S.

- Mapejats a memòria: ocupen espai de memòria principal i per tant aquesta es redueix. S'utilitzen instruccions normals (MOV) per tant l'accés als perifèrics és molt ràpid. La conexió al bus de control del registre és el mateix que el bus de la memòria.
- Mapejats a l'espai E/S: no ocupen espai a meòria. L'accés es fa amb instruccions especials (IN, OUT)

i per tant l'accés es més lent. La conexió al bus de control del registre és diferent que la conexió a memória.

No tots els ordiadors poden fer la distinció entre memòria principal i espai d'E/S. Com per exemple els ordinadors basats en processador Motorola.

Cada coontrolador ha de tenir al menys els registres d'estat i el de dades.

Hi ha diversos modes per fer la conexió, es a dir, per fer el transport de dades de la CPU al perofèric i a l'inrevés. Aquests modes són els següents:

- *Mode bloqueig*: des de que la CPU demana un servei, aquesta dedica tot el temps a aquest servei fins que termina aquest servei. Per exemple si a impresora està en mode bloqueig, la CPU no fa res fins que aquesta imprimeix el que li ha manat.
- *Mode registre d'estat*: la CPU en tot moment te el control del sistema. Va mirant si pot fer altres feines mentre ha demanat un servei a algún perifèric. Aquí apareix el concepte de overhead, el qual és un mecansme mitjançant el qual sabem si un perifèric a terminat la seva feina. Per saber si hi ha overhead es va mirant periodicament el registre d'estat. Per tant ara només es perd el temps de llegir el registre d'estat i no tot el temps en fer el servei com passava amb el cas anterior.
- *Mode interrupció*: el perifèric es qui ens avisa quan acaba de fer alguna feina o quan pot acceptar noves dades (el més normal). És el métode més utilitzat en els PC's.
- *Mode canal*: sistema que preten reduir el temps d'utilització del bus. Normalment les dades fins que arriben al perifèric han d'anar, primer, de memòria a CPU i després de CPU a perifèric, per tant es passa dos cops pel mateix bus. Per evitar això existeix el DMS, el qual amb una adreça fa que el contingut vagi directament de memòria a perifèric. És el metode més utilitzat en els ordinadors grans.

Controaldors dels Perifèrics

Els controladors poden ser de tres tipus (només diferenciant la part del controlador que enllaça amb el bus del PC):

- Hardware especialitzat: aquest hard només realitza una funció i per tant si no es té s'ha de canviar forzossament, si es necessita el perifèric.
- Hardware general: aquesta ultima fase es conecta directament entre PC i controlador. El port serie, n'es un exemple, el qual és un cicuit programable i a més es pot trobar a qualsevol ordinador.
- Hard microprogramable: al final tenim unmicrocontrolador. Aquest mode és molt més flexible, és més programable i per tant pot acceptar més perifèrics, perque el podem programar al nostre gust, segons el perifèric conectat.

Els controldors han de facilitar la interconexió dels bussos i permetre que els programin (encra que només sigui una serie de paràmetres).

Port serie del PC (8250)

El port serie és asincron, en concret és el microchip 8250.

Internament te un bus únic, un buffer (que permet la comunicació bidireccional amb el bus de dades del PC, el buffer és de 8 bits), un bloc de control lògic (el qual fa operacions de llegir −read−, escriure −write−, i operacions lògiques. Aquest genera senyals que es distribuixen als blocs), també ens trobem el sistema de rellotge, el generador de frequència (genera dues freqüències multiplicades x16 de la que li arriba, un freqüènca és pel receptor de dades extern i l'altra per l'emissor), existeix el controlador d'interrupcions (aquest treu un senyal a l'exterior i rep diversos senyals interns a partir dels quals genera els senyals exteriors), trobem

un buffer de transmissió (el qual te dos registres, el registre de transmissió de 8 bits on guarda la informació i el registre de desplaçament el qual te un bit d'start, un de paritat i un d'stop).

Segons els tres bits de menys pes de les adreces que entren al control lògic, es podrá accedir a un o altre registre. Les combinacions i els registres als que es pot accedir són els següents:

A2:0 (en decimal)	**Registre que s'accedeix**	
0	Registre de transmissió, només d'escriptura	
0	Registre de recepció de dades	
0	BAL	Registre que controla la freqüència
1	BAH	
1	IER, registre d'habilitació d'interrupció (màscara)	
2	IIR, identificador d'interrupció	
3	LCR permet programar la linia de transmissió	
5	LSR, dona l'estat de la linia	
4,6	Control de l'estat del modem	

L'accés a LCR ens indica els bits d'stop i defineixen la longitud de la paraula, les diferents combinacions són les següents:

- BCL: bit 2 i indica els its d'stop.
- CL1: bit 1
- CL0: bit 0
- PM0: bit 3
- PM1: bit 4
- PM2: bit 5
- SBK: bit 6, si està a 1 indica que s'ha provocat un break a la linia, es a dir, que l'hem interrumput.
- DLAB: bit 7, si està a 1 ens indica que les adreces 0 i 1 ens permeten accedir als registres BAL i BAH.

SBL

CL1

CL0

Longitud

Bit stop

PM0

PM2

PM1

0

0

0

5

1

0

X

X

No control paritat

0

1

6

1

0

0

Control paritat senar

1

0

7

1

0

1

Control paritat parell

1

1

8

1

1

0

Control paritat alta

1

0

0

5

1.5

1

1

1

Control paritat baixa

0

1

6

2

1

0

7

1

1

8

El significat dels bits del registre d'estat LSR és el següent:

- *RxDA*: bit 0, indica que s'ha rebut alguna dada.
- *OE*: bit 1, indicaque s'ha rebut una dada però que la dada que havia abans no s'ha llegit.
- *PE*: bit 2, indica un error de partat.
- *FE*: bit 3, indica l'error framing, aquest es produeix quan s'ha rebut la paraula però hi ha hagut algún error en el transport.
- *BKD*: bit 4, indica que s'ha detectat que la lina está en break.

- *TxDE*: bit 5, indica que el registre de transmissió està buit.
- *TxST*: bit 6, indica que el registre de transmissió i el de desplaçament estàn buits.

El sistema de rellotge: si el rellotge és intern, podem posar un senyal de rellotge extern amb 18,432 Mhz com a màxim. Amb aquesta freqüència el generador de freqüència veurà una freqüència de f/10, i el sistema de rellotge una freqüència de f/2. Però si pel contrari el rellotge és extern la freqüència màxima podrà ser de 9,216 Mhz, pel sistema de rellotge li entrarà la mateixa freqüència que abans però al generador de freqüència li entrarà la freqüència f/5.

El generador de senyal de transmissió: dintre del PC tenim una freqüència de 1,8432 Mhz i al generador de senyal de transmissió li arribarà una freqüència de f'=1,8432/16, i la freqüència que sorti serà la d'entrada dividit pel contingut dels registres BAH/BAL. $f' = \frac{1,8432}{16}$

y $f = \frac{f'}{BAH / BAL}$

.

Si carreguem el valor 00C0h als registres BAH/BAL tindrem una freqüència de sortida de 9600 bauds.

Sistema d'interrupció: el port serie ens pot donar una interrupció per:

- Condició en la recepció: error de paritat, error d'overlow, break o error framing (hem llegit les daes però hi ha hagut un error en el protocol).
- Perque hem rebut una dada.
- Perque tenim el registre de transmissió disponible.
- Per condicions de modem.

Nosaltres podem controlar les interrupcions amb un registre que les emmascara, aquest registre és el IER. EN aquest registre es poden inhibir les interrupcions:

Bit 0	RxD, recepció de dades
Bit 1	TxD, trnasmissió
Bit 2	RxCondició en la recepció
Bit 3	Modem

El registre IIR ens indica quina interrupció s'ha produit:

Bit 0 (B0)	A 1 indica que hi ha interrupcions pendents		
Bits 2−1 (B2−1)	Indica d'on ve la interrupció		
	00	Modem	
		01	TxD (Transmissió)
		10	RxD (Recepció)
		11	RxCondició.

Port Paral−lel (8255)

Aquest port és programable, però amb limitacions ja que està dins del PC i per tant s'ha de modificar dins d'aquestes limitacions. Te tres ports de 8 bits, que poden programats tant com per entrada com per sortida.

També conté dos controladors, un pel port A i un altre pel port B. Un buffer on es guarda la informació i comunica el port paral−lel amb el bus intern del PC i el control lògic.

Els tres ports poden treballar en tres modes diferents:

− Mode 0: on els conroladors no fan falta ja que llegim directament el que hi ha, i el que escrivim es enviat.

- Mode 1: el port C desapareix, i es produeix una sincronització quan es reb la informació i s'envia.
- Mode 2: en aquest mode tenim bidireccionalitat al igual que en els altres cassos, però només es trablla amb el port A.

Ara veurem les condicions, per a l'entrada o sortida de dades dels diferents ports al bus del PC:

@	RD	WR	
0	0	1	A al bus
1	0	1	B al bus
2	0	1	C al bus
0	1	0	Bus a A
1	1	0	Bus a B
2	1	0	Bus a C
3	1	0	Bus al registre de control de 8 bits.

Com a registre d'estat s'utilitza el port C que donarà aquesta informació, però per això els ports no deuen ser independents entre ells, es a dir, en Mode 1 o Mode 2.

El registre de control és de 8 bits i segons el bit 7 estigui en 1 o 0 els altres bits tindràn un o altre comportament.

D7	1	No es controla els bits del port C		
	D0	Indica si el port C en la part baixa és entrada o sortida (1−E, 0−S)		
		D1	Indica si el port B és d'entrada o sortida (1−E, 0−S)	
		D2	Indica si nosaltres tenim el port B en mode 0 o 1	
		D3	Indica si el port C part alta és entrada o sortida (1−E, 0−S)	
		D4	Indica si el port A és d'entrada o sortida (1−E, 0−S)	
		D5−6	Indica el mode de funcionament del port A:	
				Mode 0 − 00 Mode 1 − 01 Mode 2 − 10
D7	0	Controlem els bits del port C		
	D0	Indiquem si el bit indicat als bits 3−1 es posarà a 0		

	o a 1	
	D3−1	Indiquem el bit a modificar.

Ara veurem amb esquemes els diferents modes de funcionament del port parl−lel:

STB: quan el port A te una dada aquest senyal es posa a 0 i memoritza la dada.

IBF: inidica que ja s'ha memoritzat la dada, o que el port A està ocupat.

INT: genera una interrupció.

RD: inidica que llegim.

OBF: inidca que hi ha una nova dada al port port A per ser emesa.

ACK: notifica que s'ha llegit la dada.

INTR: interrupció dient que el port està lliure.

Arquitectura

Els drivers per nosaltres seràn els únics que programaran el hardware. Amb un programa d'aplicació no podrem tocar el harware i per tant necessitarem unes funcions per a que es modifiqui aquest hardware i degut a això només farem crides a aquestes funcions. Nosaltres cridem a aquestes funcions amb CALL o INT en llenguatge ensamblador.

La funció que te que consultar dades, ha de mantenir les interrupcions inhibides fins al final de la consulta, ja que podria arribar una nova interrupció y modificar la dada que estaba llegint, per tant el temps de consulta s'ha de reduir al mínim. En canvi la rutina de tractament de les interrupcions si ha de permetre les inetrrupcions, a no ser que hagi de llegir dades.

Entre aquestes funcions ha d'haver una rutina de sincronització (RSI), ja que el perifèric és més lent que la CPU. Aquesta RSI fa el desplaçament de la informació dels registres dels controladors a les nostres dades i viceversa. Aquesta RSI es qui toca el hardware juntament amb les funcions descrites.

Una altra de les funcions és la inicialització del hardware. Les funcions són particulars de cada perifèric però han de ser el més genèriques possibles, per així fer pocs canvis quan es canvia de perifèric. Aquestes funcions es troben a memòria i es carreguen juntament amb el sistema operatiu.

Per diferents perifèrics hi haurà diferents funcions però pot tenir el mateix buffer, on s'emmagatzema la informació. Quan es modifiquen les dades cal inhibir les interrupcions i quan s'executa una RSI, cal haver deixat les dades en un format coherent, per a que si ve alguna interrupció no esborri la informació de la interrupció anterior. Quan s'escriu al buffer s'ha d'escriure una dada completa, així per tant en un port serie s'ha d'esperar a que arribi la dada completa abans d'escriure−la.

Exemple: Implementació d'un subsistema d'E/S

Mecanisme de bloqueig.

Amb aquest mecanisme es consulta el registre sempre mirant si te una nova dada o per escribir−hi alguna, mentre es demana una acció a un perifèric. Per tant en l'aplicació es farà un bucle mentre el perifèric estigui fent la seva feina. Els requisits per a aquest mecanisme són els següents:

- Cal inhibir la RSI de sincronització: ja que sempre es va consultant el registre i no és el perifèric qui ens avisa, per tant no necessitem les interrupcions, sempre atenem al perifèric.
- Cal definir una funció per llegir el registre d'estat, ja que s'ha de fer la sincronització i com no tenim la RSI, hem d'anar mirant el registre d'estat però com el registre és hardware, no ho pot mirar l'aplicació sino una fucnió.
- Cal definir una altra funció per fer la transferència de dades.

Mecanisme de consulta del registre d'estat.

En aquest mecanisme, cada cert temps es consulta el controlador per si s'ha de fer alguna transferència o no. El requisits són:

- Cal inhibir la RSI.
- Funció per llegir el registre d'estat.
- Funció per fer la transferència.

Quan s'arriba a un punt del programa es fa la consulta amb la instrucció INT que executarà una rutina de servei de perifèric que mirarà el registre d'estat. Si no s'ha de fer transferència continuem amb l'aplicació i en un altre punt es torna a mirar el registre d'estat. Però és molt difícil calcular els llocs on posarà cada instrucció INT de consulta, per facilitar aquesta tasca existeix l'interval−timer el qual es pot programar per donar un senyal cada cert temps. Per tant, i amb l'ajut de l'interval−timer una altra opció d'aquest mecanisme seria:

- Definir una RSI que controli l'interval−timer, aquesta RSI agafarà la dada del registre d'estat, mirarà un bit i en funció d'aquest farà o no la transferència.

Mecanisme d'interrupció.

Els requisits del mecanisme d'interrupcions són els següents:

- Cal tenir una RSI.
- S'ha de tneir un controlador que generarà les interrupcions.
- La unitat de procés ha de tenir una entrada d'interrupcions (interrupcions mascarables i no mascarables, per tant seràn dues linees).

Al haver més d'un controlador necessitem un mecanisme per a que només hi surti un senyal d'interrupció que serà el que entri a la unitat de procès.

El controlador de interrupcions és programat per la CPU i així la CPU li pot dir que no sigui interrumpuda per cap interrupció o que si pot sr interrumpuda (inhibir o desinhibir totes les interrupcions), o que sigui interrumpuda per algunes determinades interrupcions i per les altres no (inhbició d'algunes interrupcions).

Per tant, necessitem u controlador d'interrupcions on les seves tasques són les següents:

- Arbitratge de les interrupcions en funció de la informació de l aCPU.
- Ha de permetre emmascarar interrupcions.
- Identificació de la informació per la CPU per saber quina interrupció és. Per això ha de generar l'identificador de la interrupció.

Mecanisme Canal.

Hem de transferiri dades entre l'espai de memòria i l'espai d'E/S, es a dir, entre memòria i el perifèric (disc, disquettes, etc.). La informaió pot passar per la CPU, tant si va de memòria a perifèric o de perifèric a

memòria. Les dades del bus d'adreces són diferents per memòria i pel perifèric i per tant per a que pasi la informació per la CPU, necessitem més temps per calcular l'adreça de memòria i l'adreça del perifèric.

Existeix la posibilitat de tenir un DMA (Dirct Memory Acces), però tampos es pot utilitzar el mateix bus d'adreces per la memòria i pel perifèric, degut a això hi ha linees adicionals d'adreces on cada linia és una adreça completa que li diu al registre del perifèric que la dada és per ell. El DMA és controlat per la CPU i per tant ha de demanar a la CPU la utilització del bus.

El DMA te el seu propi clock intern i per tant si aquest rellotge és lent, no ens interesarà utilitzar el DMA sinò la CPU, per tant el DMA ha ser bastant ràpid.

El mode canal necessita:

- Una funció d'inicialització de DMA que indicará el mode de transferència i de quina posició de memòria a quina altra va la informació.
- Una funció que faci el control d'errors (si el DMA provoca un error es fa una interrupció software).

Per tant en aquest mode no utilitzarem ninguna RSI perque la sincronització la fa el DMA.

Quan inhibim la RSI, volem dir que aquesta RSI no farà la transferència de dades, ni la sincronització, sino que és un altre sistema.

3.−PERIFÈRICS D'ENTRADA

Introducció

Al PC tenim el port serie i el paral−lel a on podem conectar diversos perifèrics. Tots dos són bidireccionals per tant pot entrar informació a través d'ells. Al PC per tant tenim tres portes d'entrada d'informació: el port serie, el port paral−lel i les targes internes (red, modem intern, etc.).

Al introduir informació s'ha de sincronitzar el PC amb el dispositiu. (bloqueig, registre d'estat, interrupcions, canal). El port serie en mode bloqueig és molt lent ja que tracta bit a bit.

A continuació veurem els diferents tipus de sincronització recomenables pels dos ports:

	Port Serie	Port Paral−lel
Bloqueig	−−−	Si
Reg. Estat	Si	Si
Interrupcions	Si	Si
Canal	Si	Si

La gestió del port serie mitjançant el registre d'estat necessita quatre funcions. Aquest port utilitza la interrupció 14. Les funcions que utilitza són les següents: inicialitzar, enviar carácter, llegir carácter i consultar el registre d'estat.

Quan llegim una dada del registre de dades del port srie, aquesta dada va diectament a un rgeistre de la CPU. En canvi si la gestió es fés amb interrupcions la informació va a un buffer abans de donar el resultat a l'aplicació. Amb interrupcions necessitariem una RSI, i les funcions de inicalització, d'obtenir dada, d'escriure dada i de consulta del registre d'estat.

El port paral−lel està més dissenyat per conectar només la impresora encara que es podría utilitzar per

conectar altres dispositius.

Ratolí

El ratolí es conecta al prt serie del PC. Hi ha dos formats de ratolins, per Microsoft ha definit l'estandar que s'utiltza avui en dia. Aquest ratolí te dos botons, està conectat al port serie i es gestiona per interrupcions.

Els dos tipus de ratolins són:

- Els conectats al port serie.
- Els conectats directament al bus, on es necessita una placa interna i la comunicació entre PC i ratolí es realitza, igualment, en serie.

L'estructura de la controladora és la següent:

Segons com s'obtenen les dades del moviment del ratolí, aquests es clasifiquen en:

- Mecànic: el moviment es detecta mitjançant el moviment dels rodets produit pel moviment de la bola.
- Optomecànic: te sensors enlloc de rodets, pero continua tenin l abola.
- Optic: te detectors que al vellugar−se el ratolí sobre una taula especial dona els polsos de moviment, per tant no te rodets ni bola.

Comunicació del ratolí amb el PC

El ratolí envia tres paraules consecutives per enviar totes les dades.

La velocitat de comunicació és de 1200 bauds, el ratolí envia les paraules de 7 bits sense paritat, i les dades són enviades, normalment, cada 0.01s pel ratolí.

1ª paraula

B6: primer bit (bit 6, bit de mes pes) sempre ha de ser 1, i serveix per sincronitzar.

B5−B4: dona informació si algú boto ha estat pitjat. El B2 a 1 indica que el botó esquerra a estat pitjat, i el B3 a 1 inidica que el botó dret ha estat pitjat.

B3−B2: indica si el moviment produit és en l'eix Y.

B1−B0: indica si el moviment produit és en l'eix X.

2ª paraula

B6: sempre a 0 i s'utilitza per la sincronització.

B5−B0: sis bits que indiquen quin és el desplaçament en l'eix X.

3ª paraula

B6: sempre a 0 i s'utilitza per la sincronització.

B5−B0: sis bits que inidquen quin és el desplaçament en l'eix Y.

El ratolí no ha de ser tan precis com el teclat, ja que el ratolí si no arriba a on volem el podem tornar a moure, i per tant haurá de ser precis el programa que mostri el cursor per pantalla (al mateix moment que s'obtenen les dades del ratolí s'ha d'actualitzar la opsició del cursor a pantalla, per tant l'actualització del cursor es fa al mateix moment que quan s'obtenen les dades del ratolí).

La conexió del ratolí al PC es fa amb la rutina de servei de perifèric de l'adreça 33h. Per tant si es vol executar alguna funció del ratolí s'ha de moure al registre AX el número de funció a executar i despres fer la crida.

MOV AX <− número_funció

INT 33h

Algunes de les funcions implementades pel ratolí són les següents:

- 0h fa el reset.
- 1h mostra el cursor a pantalla.
- 2h amagar el cursor.
- 3h obté la posició i estat dels botons.
- 4h moure el cursor.
- 5h obtenir el número d'activacions dels botons en una posició.
- ?? limitar la zona de desplaçament.
- 0Fh fixar la relació entre mikeys i punts.

La unitat de longitud del ratolí és el mikey (unitat de longitud de desplaçament del ratolí fins que es genera un pols).

1 mikey= $\frac{1}{200}$

polsades= 0.127 mm 1 polsada= 200 polsos.

Per realitzar la visulaització del cursor, es disposa de dues mascares, que permeten mostra el cursor com volguem.

Teclat

El teclat dels PC−XT tenen 83 tecles i els teclats dels PC−AT tenen 101/102 tecles.

El teclat te un microcomputadorque es comunica amb el PC amb una linea serie i al PC esta el controlador (I−8042). Només hi ha comunicació des de teclat a PC. El microcomputador (8048) genera una interrupció cada vegada que es pitja una tecla. El codi de la tecla no és el codi ASCII del carácter, sino el carácter d'escombrat (Scan Code), que es codi que representa la posició de la tecla pitjada al teclat.

El teclat envia una paraula de 7 bits (Scan Code) i un bit mes que serveix per indicar si una teclat ha estat pitjada (Make) o s'ha deixat anar (Break), això permet diferenciar les combinacions de tecles.

Aquestes dades s'emmagatzemen en un buffer del PC, cada tecla ocupa dos espais (s'ha de guardar el make i break de cada tecla).

En la part de sitema tindrem:

- Una RSI que llegeix la dada del teclat i la guarda a un buffer circular.

- Una rutina de servei al tecllat que permeterà consultar les dades d'aquest buffer. Les funcions que tenim són:

Funció 0: obte el codi del buffer, si el buffer esta buit es bloqueja esperant fins que arribi alguna dada. També elimina el codi llegir del buffer.

Funció 1: consulta si el buffer esta buit. Retorna el codi del caràcter, però no l'elimina del buffer.

4.− PERIFÈRICS DE SORTIDA

En l'actualitat hi ha molt programes d'aplicació i moltes impresores i pantalles. I cada impresora te el seu llenguatge de comunicació. Per establir la comunicació existeixen els drivers, amb els quals tindriem un driver per cada tipus d'impresora i programa per tant el volum dels programes seria molt gran.

Ara es considera un punt intermig (es fa una imatge del que es vol imprimir o veure per pantalla), així tots els programes haurien de posar la informació a la imatge i es d'aquí d'on s'imprimeix, on des d'on surt per pantalla.

Pel que fa a la impresora el que interessa és que ens deixi el mes aviat possible lliure per continuar fent treballar la CPU, en canvi en la pantalla interessa que s'actualitzi el més ràpid possible.

Amb la imatge creada per l'aplicació, es tindran funcions de la rutina de servei de perifèric que l'agafarà i la possarà al seu buffer.

Quan hi ha moltes dades a imprimir s'utilitza aquest mètode.

Impresora

La comuniació de la impresora amb l'ordinador (PC) es fa mitjançant un cable gruix i curt que es conecta al port paral−lel. La comunicació es realitza amb el cable centronics.

Registre de dades: és bidireccional, i el seu desplaçament respecte l'adreça base es 00h. És de 8 bits i el contingut és la dada 0 és al bit 0, la dada 1 al bit 1 i així respectivament.

Registre d'estat: el registre és de lectura, i el offset que te és 01h.

Bit 0 −Bit 2 :	Reservats
Bit 3:	ERR
Bit 4:	ONOFF
Bit 5:	PAP
Bit 6:	ACK
Bit 7:	BSY

Registre de control: aquest registre és bidireccional però que en realitat només s'utilitza d'escriptura, i el seu offset és 02h.

Bit 0:	STR
Bit 1:	ALF
Bit 2:	INI
Bit 3:	DSL

Bit 4:	IRQ
Bit 5−Bit 7:	reservats

Tots els senyals es poden trobar dins del cable centronics, excepte el senyal IRQ que permet inhibir les interrupcions. (Si IRQ està a 0 les interrupcions estaràn inhibides, es a dir, no permeses).

Avui en dia podem trobar dos tipus de conectors del port paral−lel, un conector amb 25 pins o un altre amb 36 pins.

El següent quadra resumeix el significat dels senyals i el pin on es troben:

Nº Pin				
25 pins	**36 pins**	**Senyal**	**I/O**	**Informació**
1	1	STR	Output	Si fem transferència en flag negatiu (0 −> 1) la impresora captura la dada.
2−9	2−9	D0−D7	I/O	Dades
10	10	ACK	Input	0 −> resposta de impresora dient que ha rebut una dada i està llesta per rebre més.
11	11	BSY	Input	0 −> bit que indica que la impresora està ocupada (buffer ple, impresora fora de linia o error d'impresió).
12	12	PAP	Input	1 −> a la impresora li falta paper.
13	13	ONOFF	Input	1 −> la impresora està en funcionament.
14	14	ALF	Output	0 −> la propia impresora es generarà el senyal de fi de linia. 1 −> S'ha d'indicar el fi de linia.
15	32	ERR	Input	0 −> error en la impresora (fora de linia, sense paper, altre error).
16	31	INI	Output	0 −> indica que la impresora faci una inicialització, després s'ha de posar a 1.
17	36	DSL	Output	0 −> li diem a la impresora que està relacionada amb al port paral−lel.

Adreces

Hi ha PC's que només reconeixen 2 port paral−lels, pero avui en dia quasi tots reconeixen 4 port paral−lels.

Les adreces de la bios que contenen les adreces físiques dels diferents ports parl−lels són:

- 4008h (word) indica l'adreça base de LPT1.
- 400Ah (word) indica l'adreça base deLPT2.
- 400Ch (word) indica l'adreça base de LPT3.
- 400Eh (word) indica l'adreça base de LPT4.
- 4011h (byte) indica el nº d'impresores instal−lades (nº de ports paral−lels instal−lats).
- 4078h (byte) indica els segons per timeout (si en aquest temps no arriba dada es desconecta) per LPT1.
- .
- 407Bh (bytes) indica els segons timeout per LPT4.

Les impresores normalment estaràn conectades per pollgin (registre d'estat) i no per interrupció encara que es

podria fer.

En el PC tenim dues crides per enviar informació a través del port parl−lel, aquestes dues crides són INT 21 i INT 17. Aquesta última permet accedir a 3 funcions:

- 0 −> enviem carácter en ASCII al port.
- 1 −> incialitzar l aimpresora.
- 2 −> retorna l'estat de la impresora, el codi que retorna és el mateix que el del registre d'estat més un bit (bit 0) que indica el timeout.

Pantalla

A la pantalla li fem escombrats per posar els caràcters. Aquest escombrat és lineal i s'utilitza per enviar la informació a la pantalla en serie.

La pantalla es divideix en scan−lines i cada una d'aquestes linies tindrà el seu número de pixels. A la pantalla enviem informació i entre punt i punt (del límit esquerra al límit dret) es defineix la línia de pantalla , després de cada límia es produeix el sincronismes de linia que indica que es canvi de línia. Quan s'arriba al final de la última línia hi ha un altre senyal, el sincronismes de quadre.

Per tant per escombrar la pantalla hem d'indicar 3 coses:

- Informació del punt.
- Sincronisme de línia.
- Sincronisme de quadre.

Els formats més usuals de les pantalles monocromes són (carácter x linia): 80x25, 40x25, 50x32.

Per posar un carácter es necessiten un nombre determinat de scan−lines i de columnes de pixels. La representació dels caracters es pot fer amb (columna x linia) 5x7, 7x9, 7x12, 9x14.

Pas de memoria a pantalla

Per fer el traspas d'informació necessitem un senyal de rellotge que indica l'amplada dels pixels (cada pols es defineix un pixel). També tenim una memoria RAM que contindrà la inforamció de la pantalla i hem d'implementar el mecanisme l'escombrat.

Al Clk li afegim un divisor per 9 (longitud del carácter en polsos), llavors cada vegada que fa un carácter (en una sola scan−line) dona un impols. Si li afegim un comptador modul 80 (nº de caracters per línia) donaria la coordenada X del carcater que volem escriure. Del comptador surt el senyal carry out (senyal que s'activa quan arriba a 80) i aquest senyal introduit en un comptador modul 14 (nº de línies del carácter) dona la scan−line del carácter. El carry out d'aquest altre comptador s'introdueix en un altre comptador modul 25 que donaria la coordenada Y.

La informació al surtir de la RAM es fica en una ROM on tenim la matriu de punts, d'on surten 9 bits corresponents a la dada de 9 quadradets d'una scan−line. El registre de desplaçament converteix la informació paral−lela en informació serie, i es carrega cada vegada que hi ha un canvi de carácter. Cambien de carácter cada vegada que canviem el comptador modul 80 (cada 9 polsos, quadradets per carácter i scan−line).

La freqüència que utilitza el PC per la pantalla és:

- 16,257 Mhz per la freqüència del rellotge associar al pixel.

- 18,432 Khz pel rellotge linia.
- 50 Hz pel rellotge quadre.

Ara si dividem la freqüència del rellotge dels pixels per la del rellotge de la linia ens donarà com a resultat els pixels que tenim per línia (882 pixels/linia). Pero també podem calcular els pixels multiplicant els caracters que pot contenir una linia pels pixels que ocupa cada carácter, i si fem aquesta multiplicació ens dona 720 pixels/linia. Como podem observar hi han 162 pixels de diferencia entre els dos resultats, que teoricament hauríen de ser iguals, pero el que passa es que els temps d'quests 162 pixels consisteix en el temps que triga en fer−se el retorn de la linia per començar una altra linia, aquest temps és el temps perdut en el sincronisme de línia.

A la pantalla podem tenir 25 caracters, posats verticalment, i sabent que un carácter ocupa 14 linies d'altura, podem deduir que el nombre de línies per pantalla és: 15*14= 350 línies per quadre. Aquest resultat s'hauría de poder calcular fent la divisió de la freqüència del rellotge de línia entre la del rellotge de quadre donat 369 línies/quadre. La diferència ara correspon al sincronisme de quadre i per tant del temps de retorn al principi de quadre.

La informació va a la meitat de la velocitat que la velocitat dels pixels (16,257 Mhz). A la pantalla, el pixel serà negre o blanc, i pe tant va cambiant, i degut a això la freqüència mínima serà la meitat de la freqüència dels pixels, en canvi la freqüència màxima a la que pot anar seria la mateixa que la freqüència dels pixels, i consistiria en que es dibuixa una linia tota blanca o negra a la pantalla (tots els pixels del mateix color).

El temps mencionat abans dels sincronismes no s'aprofit, per tant es temps perdut.

A 16,257 Mhz hem de treure un bit (valor del pixel, iluminat o no) per tant accedim a la memòria cada vegada que s'escriuen 9 pixels (1 byte un pixel més un bit que pren valor de l'últim bit del byte).

$$\frac{16{,}257Mhz}{9}$$

= 1,806 Mhz Mhz 553 ns. Per tant cada 553 ns accedim a memòria, i per tant amb una memoria de velocitat 200 ns tindriem suficient.

Els caracters estan composats per dos bytes, el byte corresponent al carácter propiament dit, i l'atribut d'aquest.

Pantalles gràfiques

Per controlar cada punt en una pantalla gràfica, la informació ja s'ha de trobar a memòria . La pantalla monocroma només necesita 4K per guardar tots els caracters (un carácter es composa del carcater i del atribut el que ocupa un byte cada component).

Ara tenim les 720 columnes per 350 linias, pero cada pixel es blanc o negre, i enlloc de representar caracters anem representant els pixels, i cada pixel o punt te la seva propia informació (1 byte).

Per tant al tenir 720 columnes necessitarem 90 bytes per guardar només una linia de pixels de la pantalla, i tenim 350 linies, per tant necessitarem 31500 bytes (32 K) per representar una pantalla sencera, el que significa que hem augmentat en 16 el tamany de la RAM, i per tant necessitarem més velocitat a la memòria.

Si volem color, aquest color serà diferent per cada pixel, per tant a més del byte de informació del pixel necessitarem mes espai per guardar el color (per exemple si volem 16 colors necesitarem 4 bits més). Per tant ara els punts els codifiquem en plans tants com bits necessitem, i amb l'accés a un pixel o punt farem un accés a cada pla paral−lel, en la posició corresponent al pixel, de forma paral−lela.

Les palates són els llocs on guardem els colors, i si només volem una part dels colors, només necessitarem una RAM més, molt ràpida d'on surtiran 8 bits, i a la RAM es posarà a la posició 0 el color que corresponia a aquest color en la paleta superior i així amb els colors que volem, i necessitem.

En les pantalles gràfiques quan posem una dada a la memòria posem el primer bit al pla 0, el segon al pla 1 , etc..

Una altra forma de guardar la informació és en pàgines, d'aquesta forma tenim guardades diversos cops una mateixa pantalla, així al fer la regeneració de pantalla només cambien de pla, i d'aquesta forma es fa més ràpid. Posem tots els plans paral−lels, i per tant en tots els plans es guarda la informació de cop. Nosaltres habilitarem o deshabilitarem el permis pe escriure a algun pla determinat.

La funció d'aquests plans es donar rapidessa, ja que la pantalla sempre esta llegint de la memòria i escrivint−li.

Un dels chips utilitzats en pantalles gràfiques és el 6845, el qual feia les funcions del CRT, aquest chip ja està en desús, oerò era el que controlava les tarjetes CGA, EGA, etc, es a dir, les plaques abans de la VGA. Tenia 18 registres i cada registre te una @ interna, per tant s'havia d'indicar un index i despres la dada. No permetia la identificació de la placa perque no es podia llegir la configuració d'aquest chip.

El diferents modes de video que pot utilitzar una tarjeta gràfica són:

- *No entrellaçats*: els monitors primer fan unes linees i despres les linees intermitjes entre les primeres linees, per tant tenim dos quadres (el auqdre senar i el quadre parell). En un mode no entrellaçat representariem la informació d'un sol quadre, que provoca menys resolució, però també augmenta la velocitat.
- *Entrellaçat*: si utiltzem la informació dels dos quadres, parlem del mode entrellaçat i al utiltzar els dos quadres obtenim més resolució.
- *Entrellaçat/Video*: quan posem les linees parells d'un qudre fins a la meitat, i despres les mateixes linees parells del mateix quadre, i despres comencem a les linees senars de l'altre quadre posant−les entre mig de les altres, llavors parlem del mode entrellaçat/video.

Resolucions

MDA	**CGA**	**HGC**	**EGA**	**VGA**	**SVGA**