
Architettura del sottosistema di Input/Output

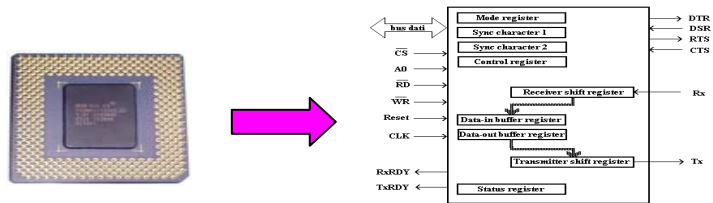
Corso di Calcolatori Elettronici I

Dipartimento di Informatica e Sistemistica
Università degli Studi di Napoli “Federico II”

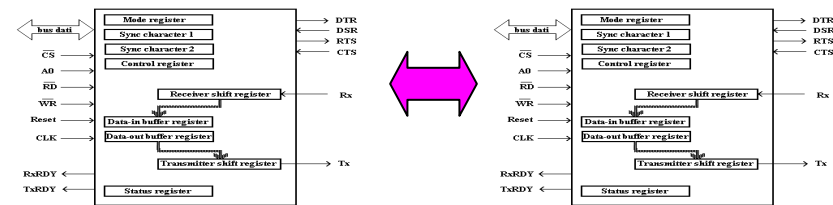


Interfacciamento di Sistemi a Microprocessore

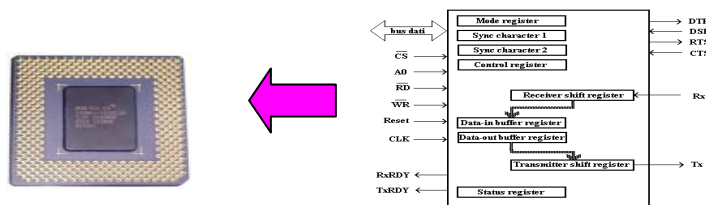
Interfacciamento Processore/Dispositivo



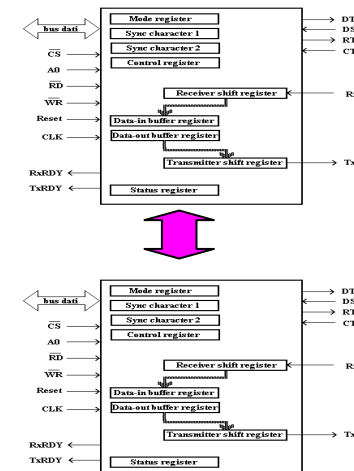
Interfacciamento Dispositivo/Dispositivo



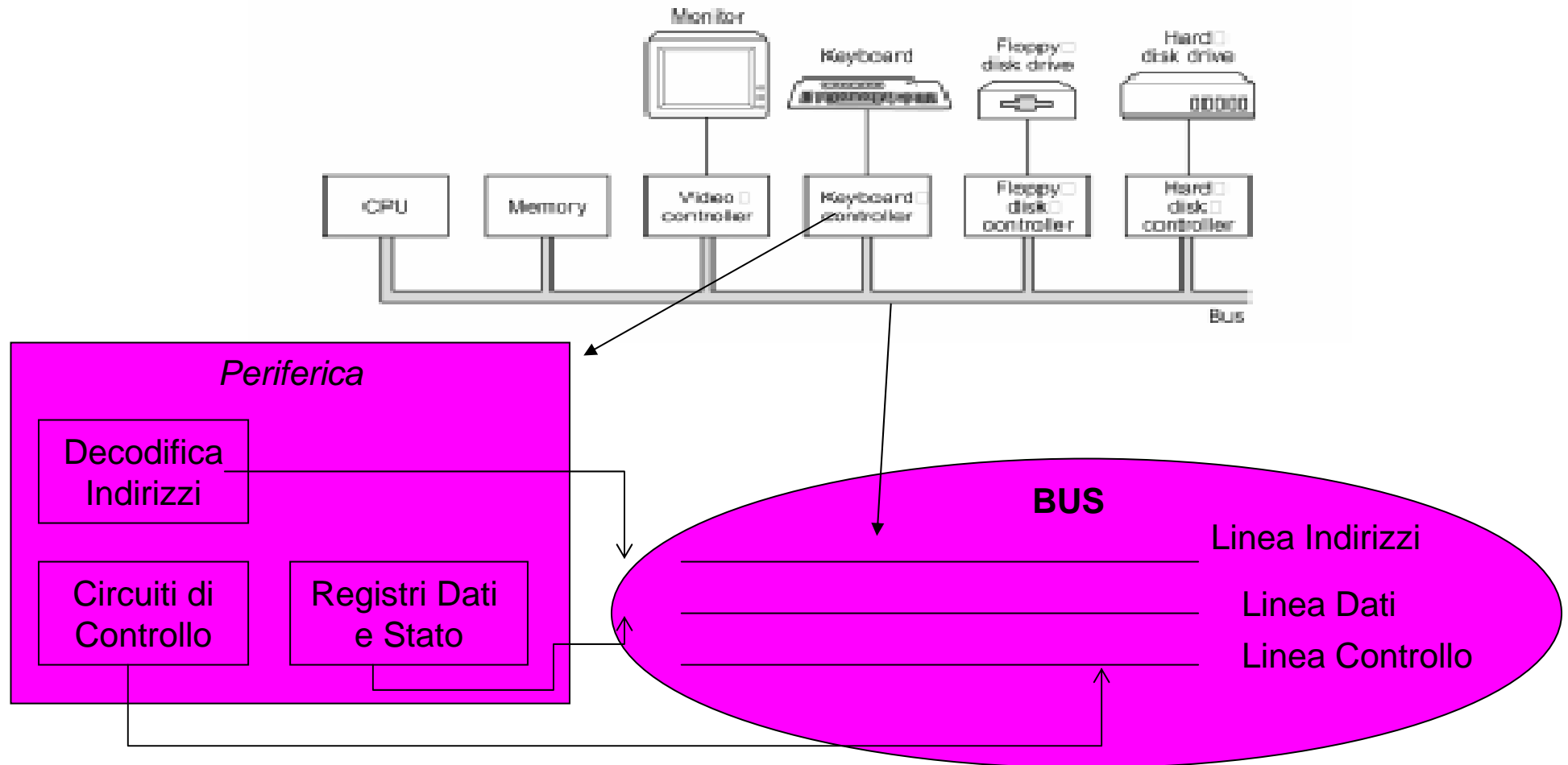
Interfacciamento Dispositivo/Processore



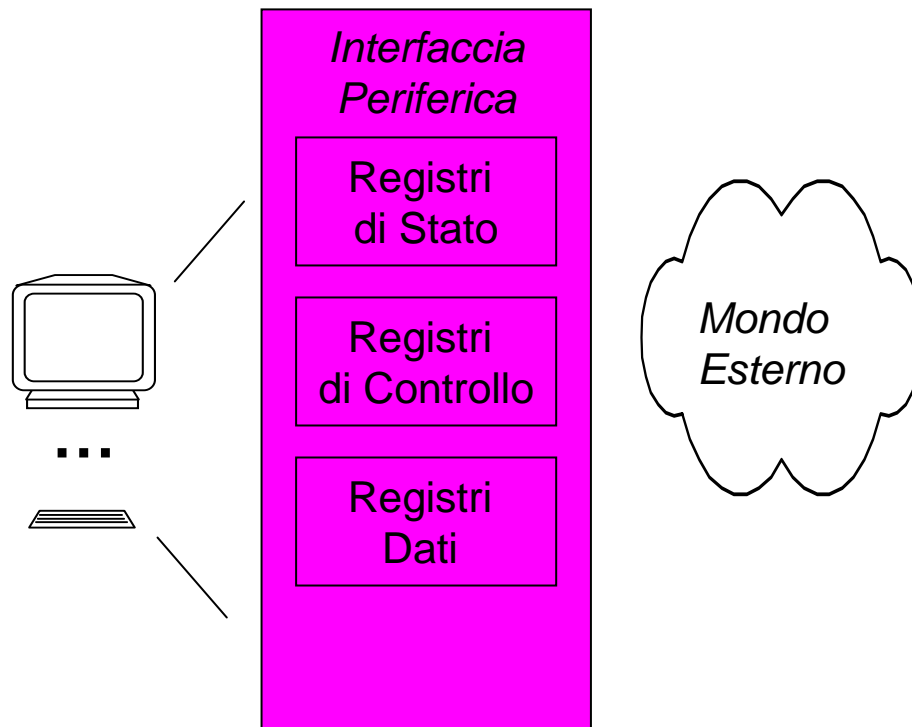
Interfacciamento di Sistemi



Il Calcolatore e le periferiche



Interfaccia



E' il modo in cui la periferica si presenta all'esterno.

In genere è una collezione di registri che permettono di

- Osservare lo Stato
- Modificare lo Stato
- Fornire Dati



Protocolli di Comunicazione

Un Protocollo è quell'insieme di regole che gestiscono la comunicazione tra due entità.

Protocollo Sincrono

E' previsto un segnale di sincronizzazione (clock) che permette di gestire la temporizzazione delle comunicazioni

Protocollo Asincrono

Tutta la temporizzazione della comunicazione è gestita dal protocollo stesso attraverso lo scambio dei messaggi.



HandShake (*Protocollo Asincrono*)

Protocollo di Ingresso

- Manda la Richiesta
- Aspetta un *Ack* (Strobe-In)
- Leggi il dato

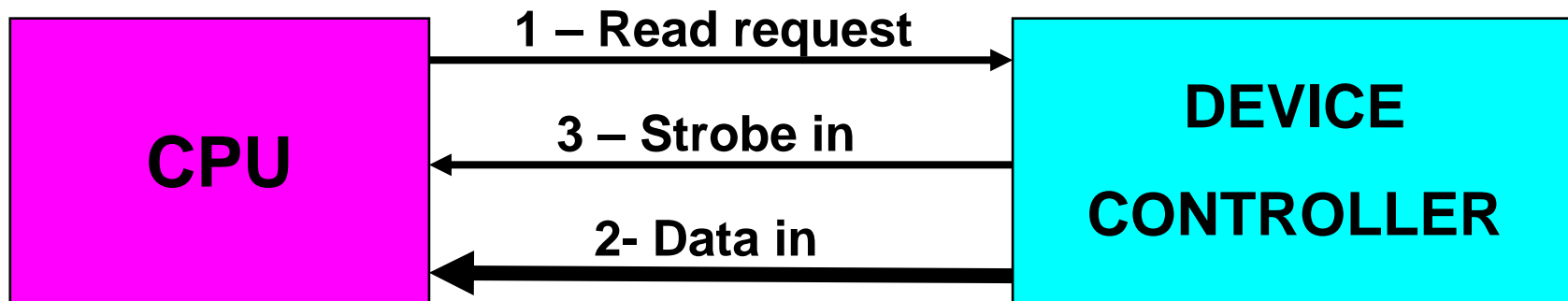
Protocollo di Uscita

- Scrivi i dati
- Manda la Richiesta (Strobe-Out)
- Aspetta un *Ack*

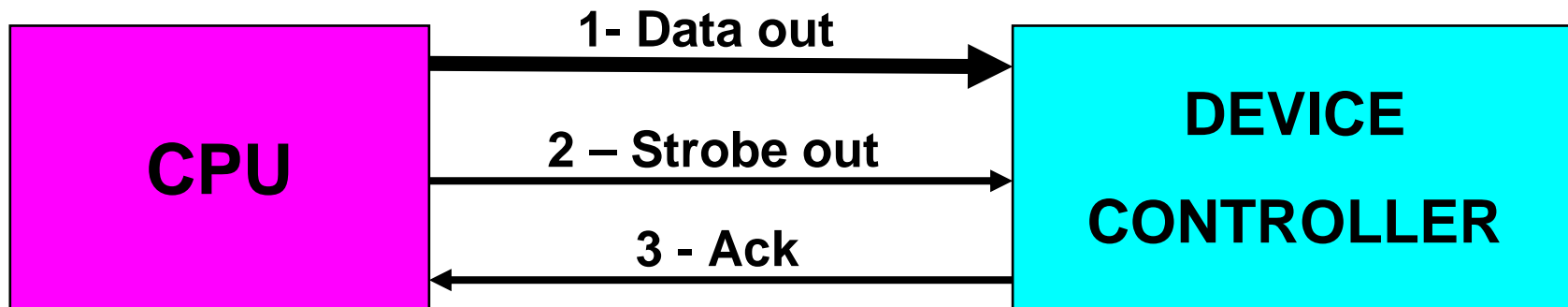


Handshake – Sequenza dei segnali

Il processore legge un dato dalla periferica



Il processore scrive un dato verso la periferica



Comunicazione Parallela e Seriale

Comunicazione Parallela

- La comunicazione avviene direttamente con una parola di un byte

Comunicazione seriale

- La comunicazione avviene un bit per volta



I problemi della comunicazione seriale (I)

- La periferica comunica con il processore attraverso una interfaccia.
- La comunicazione tra interfaccia e processore è sempre parallela
- La comunicazione tra interfaccia e periferica avviene mediante la porta seriale.
- E' necessario convertire la comunicazione da parallela a seriale



I problemi della comunicazione seriale (II)

- Il processore comunica all'interfaccia il dato ma questo non viene immediatamente comunicato
- Per comunicare un nuovo dato è necessario aspettare che termini la comunicazione precedente
- Ampio utilizzo dei buffer



Tipi di comunicazione seriale

Trasmissione Sincrona

- Viene mandata una sequenza di inizio comunicazione
- Vengono mandati i bit
- Viene mandata una sequenza di fine comunicazione
- La sincronizzazione avviene grazie ad un segnale di clock

Trasmissione Asincrona

- I bit vengono mandati uno per volta
- Prima di ciascun byte viene mandata una particolare sequenza di sincronizzazione
- Non viene utilizzato nessun clock esterno



Elementi Fondamentali della Comunicazione

- Lato Periferica
 - » Interfaccia della periferica
 - » Protocollo di Comunicazione
- Lato Processore
 - » Meccanismo di Controllo
 - » Tipo di istruzioni di I/O
 - » Driver

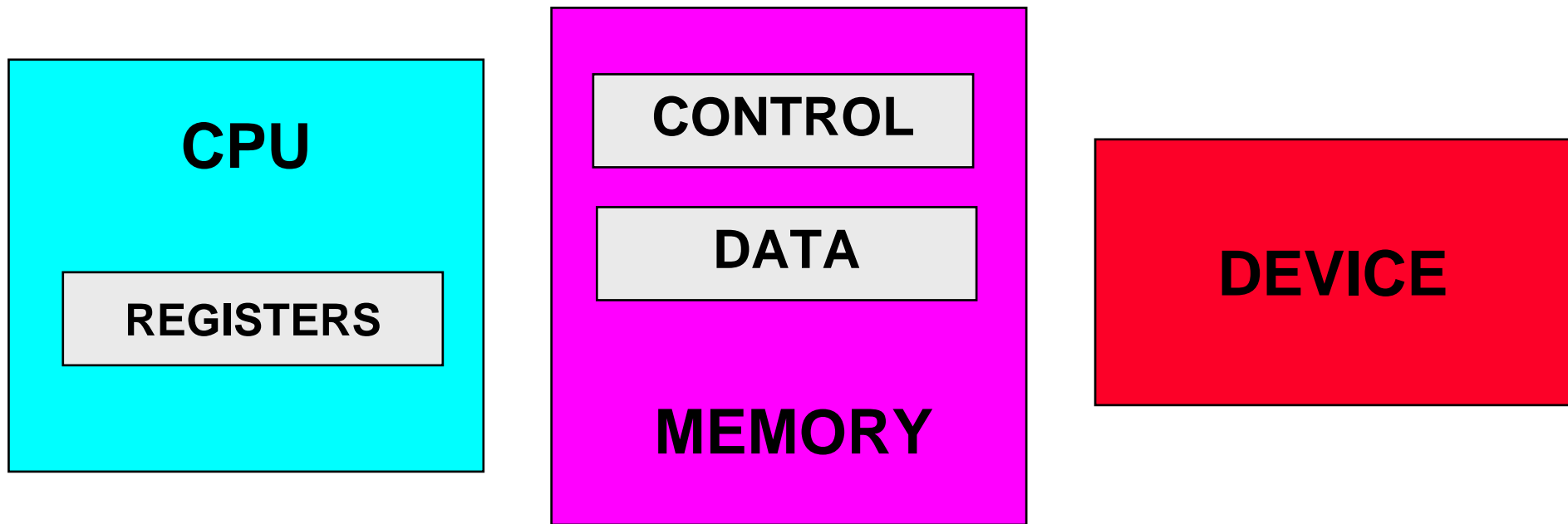


Processore ed I/O

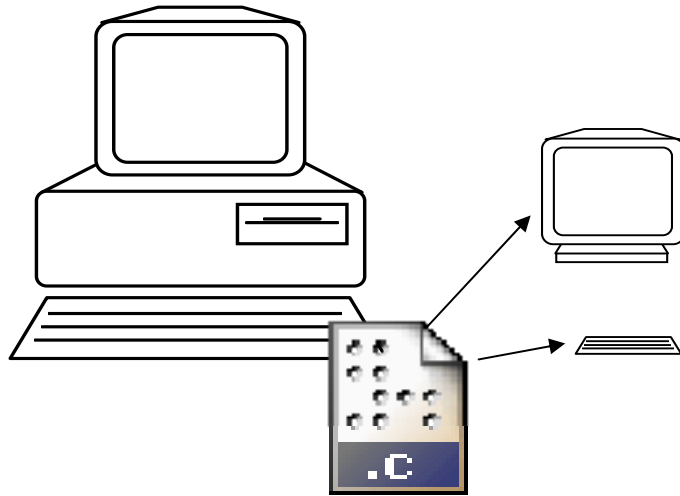
- Istruzioni di I/O
 - » Istruzioni Specifiche
 - » Indirizzamento separato per le periferiche
- Memory Mapped
 - » Lo spazio di memoria visto dal processore è condiviso tra la memoria vera e propria e le periferiche



Memory Mapped I/O - Schema concettuale



Il Driver



E' il programma che gestisce il dialogo tra il processore e la periferica.

Implementa il protocollo di comunicazione e gestisce la comunicazione

Astrae e semplifica la comunicazione tra processore e periferica

I/O Controllato da programma:

Il Driver è un programma come gli altri

Interrupt:

Il Driver è tipicamente un procedura particolare (ISR)



Meccanismi di I/O

Controllato da programma

Il Processore controlla lo stato della periferica in continuazione aspettando un cambiamento dello stato

Accesso Diretto in Memoria

La periferica è in grado di accedere alla memoria senza l'intervento del Processore

Interruzioni

Il Processore procede nel suo lavoro, quando la periferica cambia stato manda un segnale al processore che interrompe il lavoro corrente e procede a gestire l'evento.



Esempio di Driver

```
STRT  Inizializza
POLL  Testa lo stato
      BEQ POLL
      Leggi Il Dato
POLL2 Testa lo stato
      BEQ POLL
      Scrivi il Dato
      (Deve continuare?)
      BEQ POLL
```

Lo schema riprodotto mostra un esempio di comunicazione con un terminale che permette di leggere un carattere e rimetterlo a video con un meccanismo di controllo dello I/O da programma.



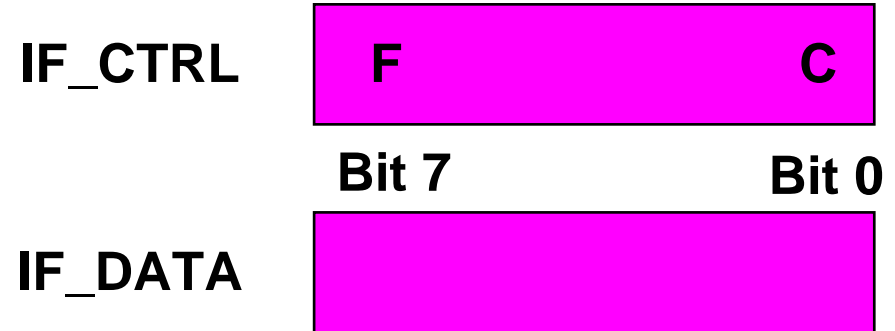
I/O controllato da programma - Output

```
DataRegister db;  
ControlBits F,C;
```

```
db.write ();  
C.set ();
```

```
do {  
  
}  
while ( F.isReset () )
```

```
C.reset ();  
F.reset ();
```



```

ORG      $8000
START    MOVE .B   D0 , IF_DATA
        ORI .B   #$01 , IF_CTRL
LOOP     BTST .B  #7 , IF_CTRL
        BEQ     LOOP
        MOVE .B  #0 , IF_CTRL

ORG      $8020
IF_CTRL  DS .B   1
ORG      $8022
IF_DATA  DS .B   1
    
```



I/O controllato da programma - Input

```
DataRegister db;
ControlBits F,C;
```

```
C.set ();
F.reset ();
```

```
do {
```

```
}
```

```
while ( F.isReset () )
```

```
db.read ();
C.reset ();
```

IF_CTRL

F

C

Bit 7

Bit 0

IF_DATA

```

                                ORG      $8000
START                          MOVE .B  #1, IF_CTRL
LOOP                            BTST .B  #7, IF_CTRL
                                BEQ      LOOP
                                MOVE .B  IF_DATA, D0
                                AND .B  #$FE, IF_CTRL

```

```

                                ORG      $8020
IF_CTRL                         DS .B   1
                                ORG      $8022
IF_DATA                         DS .B   1

```

