Instrumentación de Tiempo Real

UNIVERSIDAD DE CANTABRIA

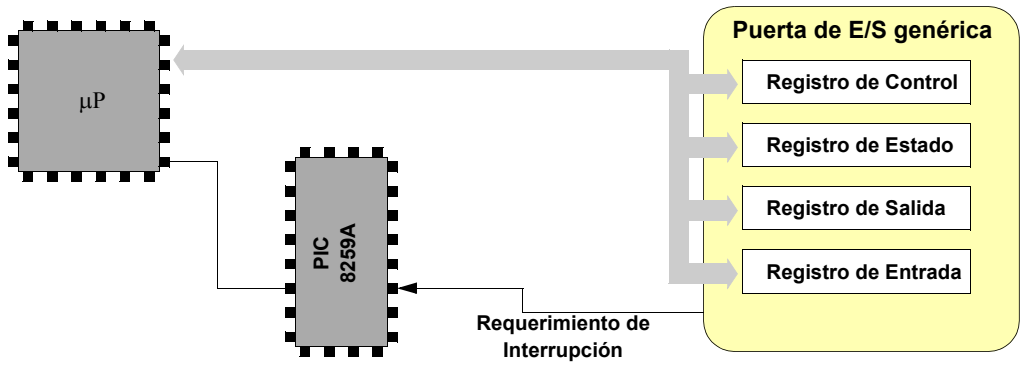
Puerta de entrada/salida genérica

UNIVERSIDAD DE CANTABRIA

Las puertas de E/S ponen al computador en contacto con su entorno:

- **convertidores A/D y D/A, entradas y salidas digitales, puertos serie y paralelo, tarjetas de red, etc.**

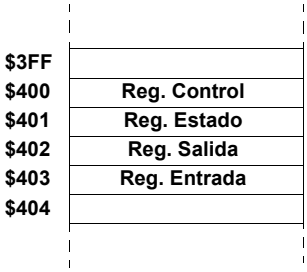
Estructura de una puerta de E/S genérica:

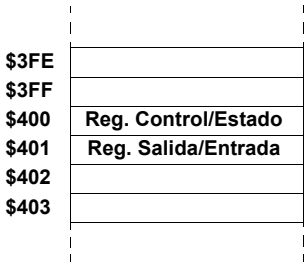


Puerta de entrada/salida genérica: Direcciones en el mapa de E/S

UNIVERSIDAD DE CANTABRIA

Cada registro ocupa una posición en el mapa de E/S





Dos registros pueden compartir una misma dirección en el mapa de E/S

- **uno de ellos será de sólo lectura y otro de sólo escritura**

A los registros se accede con las operaciones `Inb` (lectura) y `Outb` (escritura)

Puerta de entrada/salida genérica: Registro de datos

UNIVERSIDAD DE CANTABRIA

Registro a través del que el computador envía y/o recibe la información

Si la puerta es de salida, este registro es de sólo escritura

- **el programa debe escribir en él el dato que quiere transferir**

Si la puerta es de entrada, el registro es de sólo lectura

- **el programa lee el registro de datos para obtener el último dato recibido**

Si la puerta es de entrada/salida, los registros de datos de entrada y salida suelen ocupar la misma dirección de E/S

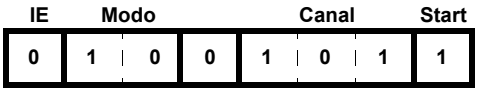
- **las operaciones de escritura son relativas al registro de salida y las de lectura al de entrada**

Puerta de entrada/salida genérica: Registro de control

UNIVERSIDAD DE CANTABRIA

Permite establecer el modo y parámetros de operación de la puerta

Ejemplo de registro de control genérico:



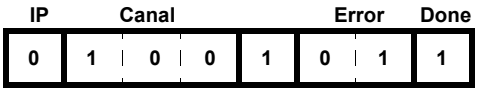
- Start: comienzo de una operación de la puerta (p.e. captura de dato)
- Canal: selecciona entre las diferentes fuentes o destinos
- Modo: modo de operación
- Habilita/inhibe interrupción (IE): controla si debe generarse una interrupción cuando ha finalizado una operación

Puerta de entrada/salida genérica: Registro de estado

UNIVERSIDAD DE CANTABRIA

Registro de sólo lectura que permite conocer el estado en que se encuentra la puerta

Ejemplo de registro de estado genérico:



- Done: indica si la puerta ha finalizado la última operación
- Error: permite conocer si se ha producido un error (y de que tipo) en la última operación de la puerta
- Canal: último canal seleccionado
- Interrupción Pendiente (IP): se ha generado una interrupción que aún no ha sido atendida

Puerta de entrada/salida paralelo

UNIVERSIDAD DE CANTABRIA

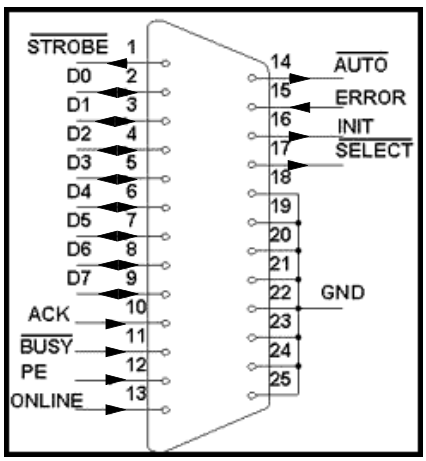
Permite trasferir información en grupos de 8 bits (un byte)

Dispone de los siguientes grupos de líneas:

- **8 líneas de datos bidireccionales:**
  - **pueden utilizarse para enviar o recibir bytes**
- **9 líneas auxiliares**
  - **originalmente pensadas para implementar el protocolo "Centronics" para comunicación computador/impresora**
  - **5 de ellas son líneas de entrada (`ACK`, `BUSY`, `PE`, `ONLINE` y `ERROR`)**
  - **4 son líneas de salida (`STROBE`, `AUTO`, `INIT` y `SELECT`)**

Puerta de entrada/salida paralelo: Conector de 25 pines

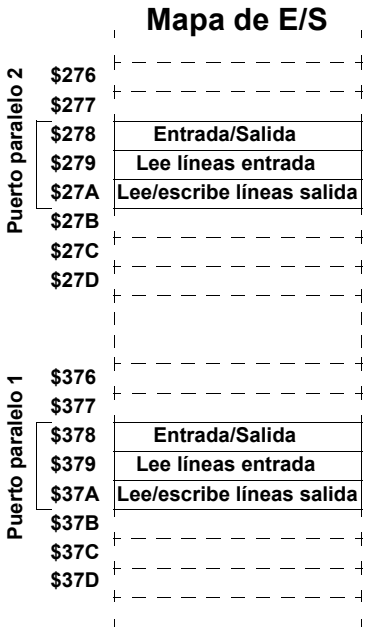
UNIVERSIDAD DE CANTABRIA



Puerta de entrada/salida paralelo: Registros del puerto paralelo

UNIVERSIDAD DE CANTABRIA

Dirección de E/S puerto1/puerto2	Registro	Modo
$378/$278	Entrada de Datos	Lect.
$378/$278	Salida de Datos	Esc.
$379/$279	Lee valor líneas auxiliares de entrada	Lect.
$37A/$27A	Lee valor líneas auxiliares de salida	Lect.
$37A/$27A	Establece valor líneas auxiliares de salida	Esc.



Registro de entrada/salida de datos

UNIVERSIDAD DE CANTABRIA

Este registro permite leer el estado de las 8 líneas de datos o establecer su valor en el caso de que la puerta actúe como salida

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
pin9	pin8	pin7	pin6	pin5	pin4	pin3	pin2

- **La puerta actúa como salida o entrada dependiendo del valor asignado al bit `Output` del registro de control**

Registro de lectura de las líneas auxiliares de entrada

UNIVERSIDAD DE CANTABRIA

Permite leer el estado de las 5 líneas auxiliares de entrada

Busy	$\overline{\text{Ack}}$	PE	OnLine	Error	X	X	X
pin11	pin10	pin12	pin13	pin15			

- **En el caso de la línea `Ack` el valor leido es el complementario al existente en el pin 10**

Registro de control

UNIVERSIDAD DE CANTABRIA

Permite leer y escribir el estado de las 4 líneas auxiliares de salida

También permite configurar el funcionamiento de la puerta:

- **puerta de salida (`Output=>0`) o de entrada (`Output=>1`)**
- **interrupción habilitada (`IRQ enable=>1`): se produce una interrupción con cada flanco de bajada en el pin 10 (`Ack`)**

X	X	$\overline{\text{Output}}$	IRQ enable	$\overline{\text{Select}}$	Init	$\overline{\text{Auto}}$	$\overline{\text{Strobe}}$
				pin17	pin16	pin14	pin1
valor tras el reset →		0	0	1	0	1	1

- **El valor de los bits `Select`, `Auto` y `Strobe` es el complementario del establecido en los pines 17, 14 y 1**

Ejemplo de uso puerta paralelo

UNIVERSIDAD DE CANTABRIA

```
with MaRTE_OS;
with MaRTE_Hardware_Interrupts;
use MaRTE_Hardware_Interrupts;
with IO_Interface; use IO_Interface;
with Basic_Integer_Types; use Basic_Integer_Types;
with System;
with Basic_Console_IO, Text_IO, Ada.Integer_Text_IO;
use Text_IO, Ada.Integer_Text_IO;

procedure Printer_Port_Test is

   -- Registros del puerto paralelo
   PP_BASE : constant IO_Port := 16#378#; -- Puerto 1
   PP_DATA_REG    : constant IO_Port := PP_BASE + 0;
   PP_STATUS_REG  : constant IO_Port := PP_BASE + 1;
   PP_CONTROL_REG : constant IO_Port := PP_BASE + 2;
```


Ejemplo de uso puerta paralelo (cont.)

UNIVERSIDAD DE CANTABRIA

```
   --  Byte leido del puerto
   Data_Read : Unsigned_8;
   pragma Volatile (Data_Read);
   New_Data : Boolean := False;
   pragma Volatile (New_Data);

   --  Manejador de interrupción del puerto paralelo
   function PP_Handler (Area : in System.Address;
                        Intr : in Hardware_Interrupt)
                       return Handler_Return_Code is
   begin
     Basic_Console_IO.Put (" In PP_Handler ");
     Basic_Console_IO.New_Line;
     Data_Read := IO_Interface.Inb (PP_DATA_REG);
     New_Data := True;
     return POSIX_INTR_HANDLED_NOTIFY;
   end PP_Handler;
```

Ejemplo de uso puerta paralelo (cont.)

UNIVERSIDAD DE CANTABRIA

```
begin
  --  Instala manejador de interrupción
  if Associate (PARALLEL1_INTERRUPT,
                PP_Handler'Unrestricted_Access,
                System.Null_Address, 0) /= 0 then
    Put_Line ("Error: Associate");
  end if;

  --  Configura como puerto de entrada y habilita las
  --  interrupciones del puerto paralelo
  IO_Interface.Outb (PP_CONTROL_REG, 2#00_1_1_0000#);

  --  Habilita la interrupción en el PIC
  if Unlock (PARALLEL1_INTERRUPT) /= 0 then
    Put_Line ("Error: Unlock");
  end if;
```


Ejemplo de uso puerta paralelo (cont.)

UNIVERSIDAD DE CANTABRIA

```
  --  Espera a que ocurra la interrupción y muestra
  --  el valor leido
  loop
    if New_Data then
      New_Data := False;
      Put ("Byte leido:");
      Put (Integer (Data_Read), Base => 16);
      New_Line;
    end if;
  end loop;

end Printer_Port_Test;
```

Tarjeta de entrada/salida AX5411: funcionamiento básico

UNIVERSIDAD DE CANTABRIA

Tarjeta de entrada/salida de propósito general

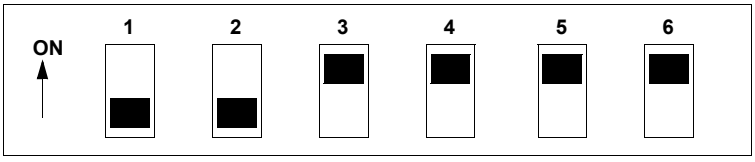
Compuesta por tres subsistemas independientes:

- Salida analógica
  - dos convertidores D/A de 12 bits
- Entrada analógica
  - 16 líneas de entrada encaminadas a un único convertidor A/D mediante un multiplexor de 16 a 1
  - máxima frecuencia de muestreo de 60 Kmuestras/seg
  - ganancia configurable
- Entrada/Salida digital
  - 24 canales de entrada y 24 de salida

Configuración hardware

UNIVERSIDAD DE CANTABRIA

La dirección base de la tarjeta en el espacio de E/S se configura mediante "switches"

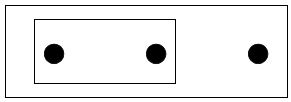


$000-00F	ON	ON	ON	ON	ON	ON
$010-01F	ON	ON	ON	ON	ON	OFF
...						
$200-20F	OFF	ON	ON	ON	ON	ON
$210-21F	OFF	ON	ON	ON	ON	OFF
...						
$300-30F	OFF	OFF	ON	ON	ON	ON
$310-31F	OFF	OFF	ON	ON	ON	OFF
...						
$3F0-3FF	OFF	OFF	OFF	OFF	OFF	OFF

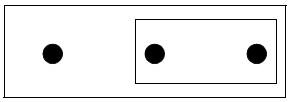
Configuración hardware (cont.)

UNIVERSIDAD DE CANTABRIA

Un puente hardware permite configurar el rango de entrada del convertidor A/D

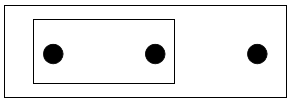


-5V a +5V

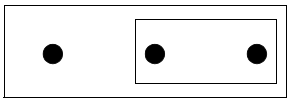


-10V a +10V

Otro puente permite configurar el rango de salida del convertidor D/A



0V a +5V



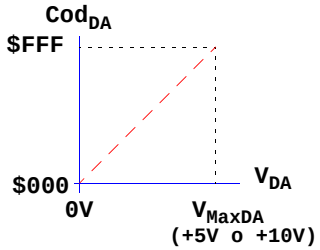
0V a +10V

Subsistema de Salida Analógica

UNIVERSIDAD DE CANTABRIA

Dos convertidores D/A:

- **Rango 0 a 5V o 0 a 10V**
- **12 bits de resolución**
- **$V_{DA} = Cod_{DA} * V_{MaxDA} / 4095$**



Registros del subsistema de salida analógica:

Dirección Base + offset	Tipo Operación	Función	Nombre corto
Base + 4 ($304)	Escritura	Low Byte del D/A 0	DAL0
Base + 5 ($305)	Escritura	High Byte del D/A 0	DAH0
Base + 6 ($306)	Escritura	Low Byte del D/A 1	DAL1
Base + 7 ($307)	Escritura	High Byte del D/A 1	DAH1

Subsistema de Salida Analógica: Registros

UNIVERSIDAD DE CANTABRIA

DAL0

Base + 4 ($304)	D3	D2	D1	D0	X	X	X	X	Byte menos significativo del convertidor D/A 0

DAH0

Base + 5 ($305)	D11	D10	D9	D8	D7	D6	D5	D4	Byte más significativo del convertidor D/A 0

DAL1

Base + 6 ($306)	D3	D2	D1	D0	X	X	X	X	Byte menos significativo del convertidor D/A 1

DAH1

Base + 7 ($307)	D11	D10	D9	D8	D7	D6	D5	D4	Byte más significativo del convertidor D/A 1

Debe escribirse siempre primero el byte menos significativo y a continuación el más significativo

Subsistema de Entrada Analógica

UNIVERSIDAD DE CANTABRIA

- 16 líneas de entrada encaminadas a un único convertidor A/D mediante un multiplexor de 16 a 1
- 12 bits de resolución
- Máxima frecuencia de muestreo de 60 Kmuestras/seg
- Ganancia configurable
- Temporizador programable integrado en la tarjeta
- Disparo (comienzo de la conversión) por software, por flanco de subida de señal externa o por temporizador interno
- Posibilidad de generar una interrupción cuando finaliza la conversión
  - tipo de interrupción configurable por software

Subsistema de Entrada Analógica: Registros

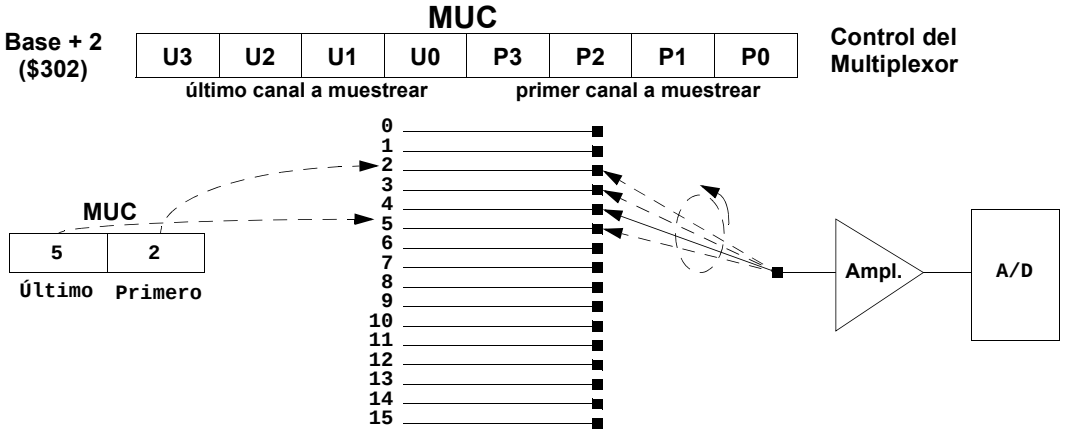
UNIVERSIDAD DE CANTABRIA

Dirección Base + offset	Tipo Operación	Función	Nombre corto
Base + 0 ($300)	Lectura Escritura	Low Byte del A/D Comienza conversión A/D	ADL START
Base + 1 ($301)	Lectura Escritura	High Byte del A/D Control de ganancia	ADH CGA
Base + 2 ($302)	Lec./Esc.	Control del multiplexor	MUC
Base + 8 ($308)	Lectura Escritura	Registro de estado Rehabilita interrupción	STATUS CLI
Base + 9 ($309)	Lec./Esc.	Registro de Control	CNTR

Subsistema de Entrada Analógica: Registro MUC

UNIVERSIDAD DE CANTABRIA

El registro *MUC* permite configurar el conjunto de canales de entrada que van a ser muestreados



- **para muestrear un único canal se pone el mismo número como primer y último canal**

Subsistema de Entrada Analógica: Registro CGA

UNIVERSIDAD DE CANTABRIA

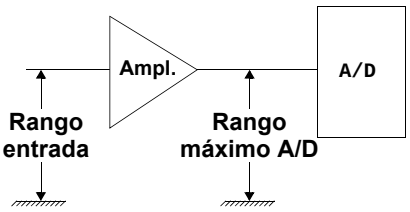
El registro *CGA* permite controlar la ganancia del amplificador colocado a la entrada del convertidor A/D

CGA

Base + 1 ($301)	X	X	X	X	R3	R2	R1	R0	Control de ganancia

Ganancia	R3	R2	R1	R0
1	0	0	0	0
2	0	0	0	1
4	0	0	1	0
8	0	0	1	1
16	0	1	0	0

$$\text{Rango entrada} = \frac{\text{Rango máximo A/D}}{\text{Ganancia}}$$



Subsistema de Entrada Analógica: Registro de control (CNTR)

UNIVERSIDAD DE CANTABRIA

CNTR

Base + 9 ($309)	INTE	I2	I1	I0	TRGE	DMAE	T1	T0	Registro de control

- ***INTE***: habilita (*1*) o deshabilita (*0*) las interrupciones
- ***I2-I1***: tipo de interrupción que generará la tarjeta
  - los valores 000 y 001 no son válidos
- ***TRGE***: se utiliza una señal externa para disparar el temporizador (*TRGE=1*) o el reloj interno (*TRGE=0*)
- ***DMAE***: habilita la transferencia de datos por DMA (*DMAE=1*)
- ***T1-T0***: disparo del convertidor A/D
  - *0 X*: comando software
  - *1 0*: flanco de subida de señal externa
  - *1 1*: temporizador interno

Subsistema de Entrada Analógica: Registro de estado (STATUS)

UNIVERSIDAD DE CANTABRIA

STATUS

Base + 8 ($308)	EOC	X	X	INTP	NC3	NC2	NC1	NC0	Registro de estado

- *EOC*: indica el final de la conversión
  - *0*: conversión finalizada, dato válido
  - *1*: conversión en marcha, dato inválido
- *INTP*: indica si hay una interrupción pendiente
  - *0*: no hay interrupción pendiente
  - *1*: hay interrupción pendiente. Permanecerá a uno hasta que reciba un comando "clear interrupt" (CLI)
- *NC3-NC0*: próximo canal a muestrear

Subsistema de Entrada Analógica: Registros START y CLI

UNIVERSIDAD DE CANTABRIA

Cualquier instrucción de escritura sobre el registro *START* provoca el comienzo de una conversión del convertidor A/D

- siempre que la tarjeta haya sido configurada para disparo por comando software

START

Base + 0 ($300)	X	X	X	X	X	X	X	X	Disparo software del convertidor A/D

Cualquier instrucción de escritura sobre el registro *CLI* rehabilita la tarjeta para generar una nueva interrupción

CLI

Base + 8 ($308)	X	X	X	X	X	X	X	X	Clear interrupt

Subsistema de Entrada Analógica: Registros ADL y ADH

UNIVERSIDAD DE CANTABRIA

Los registros *ADL* y *ADH* permiten leer el último valor convertido por el convertidor A/D

ADL

Base + 0 ($300)	D3	D2	D1	D0	X	X	X	X	4 bits menos significativos del convertidor A/D

ADH

Base + 1 ($301)	D11	D10	D9	D8	D7	D6	D5	D4	8 bits más significativos del convertidor A/D

Subsistema de Entrada/Salida digital

UNIVERSIDAD DE CANTABRIA

24 líneas de entrada y 24 de salida repartidas en 2 conectores:

- **conector principal con 8 líneas de entrada y 8 de salida**
- **conec. secundario con 16 líneas de entrada y 16 de salida**

DIN

Base + 3 ($303)	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0	líneas digitales de entrada (conector principal)

DOUT

Base + 3 ($303)	DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0	líneas digitales de salida (conector principal)

DIL

Base + 10 ($30B)	DI15	DI14	DI13	DI12	DI11	DI10	DI9	DI8	8 primeras líneas de entrada (conector secundario)

DOL

Base + 10 ($30B)	DO15	DO14	DO13	DO12	DO11	DO10	DO9	DO8	8 primeras líneas de salida (conector secundario)

DIH

Base + 11 ($30C)	DI23	DI22	DI21	DI20	DI19	DI18	DI17	DI16	8 últimas líneas de entrada (conector secundario)

DOH

Base + 11 ($30C)	DO23	DO22	DO21	DO20	DO19	DO18	DO17	DO16	8 últimas líneas de salida (conector secundario)

Ejemplo: uso de interrupciones en la tarjeta AX5411

UNIVERSIDAD DE CANTABRIA

```
with MaRTE_OS;
with Basic_Integer_Types; use Basic_Integer_Types;
with MaRTE_Hardware_Interrupts, IO_Interface;
use MaRTE_Hardware_Interrupts, IO_Interface;
with Ada.Text_IO, System; use Ada.Text_IO;

procedure Lee_AD_Con_Interrupciones is

   --  Direcciones de los registros de E/S del convertidor
   BASE_AX5411 : constant IO_Port := 16#300#;
   Reg_ADL     : constant IO_Port := BASE_AX5411 + 0;
   Reg_ADH     : constant IO_Port := BASE_AX5411 + 1;
   Reg_CGA     : constant IO_Port := BASE_AX5411 + 1;
   Reg_MUC     : constant IO_Port := BASE_AX5411 + 2;
   Reg_START   : constant IO_Port := BASE_AX5411 + 0;
   Reg_CLI     : constant IO_Port := BASE_AX5411 + 8;
   Reg_STATUS  : constant IO_Port := BASE_AX5411 + 8;
   Reg_CNTR    : constant IO_Port := BASE_AX5411 + 9;
```


Ejemplo: uso de interrupciones en la tarjeta AX5411 (cont.)

UNIVERSIDAD DE CANTABRIA

```
   --  Variables globales utilizadas por el manejador
   --  y el programa principal
   Cod : Unsigned_16;
   pragma Volatile (Cod);

   Nuevo_Dato : Boolean := False;
   pragma Volatile (Nuevo_Dato);
```

Ejemplo: uso de interrupciones en la tarjeta AX5411 (cont.)

UNIVERSIDAD DE CANTABRIA

```
   --  Manejador de la interrupción
   function Manejador_AD (Area : in System.Address;
                          Intr : in Hardware_Interrupt)
                         return Handler_Return_Code is
      Hi, Lo : Unsigned_8;
   begin
      --  Lee el código de salida del convertidor
      Hi := Inb (Reg_ADL);
      Lo := Inb (Reg_ADH);
      Cod := Unsigned_16 (Hi)*16 + Unsigned_16 (Lo)/16;
      Nuevo_Dato := True;
      --  Rehabilita interrupción
      Outb (Reg_CLI, 0);
      --  Comienza siguiente conversión
      Outb (Reg_START, 0);
      return POSIX_INTR_HANDLED_NOTIFY;
   end Manejador_AD;
```


Ejemplo: uso de interrupciones en la tarjeta AX5411 (cont.)

UNIVERSIDAD DE CANTABRIA

```
begin
   --  Configura ganancia 2 en el convertidor
   Outb (Reg_CGA, 16#01#);

   --  Muestrear sólo canal 0
   Outb (Reg_MUC, 16#00#);

   --  Instala manejador de la interrupción
   if Associate (PARALLEL2_INTERRUPT,
                 Manejador_AD'Unrestricted_Access,
                 System.Null_Address, 0) /= 0 then
      Put_Line ("Error Associate");
   end if;

   --  Configura tarjeta AX5411
   Outb (Reg_CNTR, 2#1_101_0_0_00#);
   --  INTE     I2 I1 I0     TRGE     DMAE     T1 T0
   --    1       1  0  1       0        0       0  0
```

Ejemplo: uso de interrupciones en la tarjeta AX5411 (cont.)

UNIVERSIDAD DE CANTABRIA

```
   --  Habilita interrupción en el PIC
   if Unlock (PARALLEL2_INTERRUPT) /= 0 then
      Put_Line ("Error Unlock");
   end if;
   -- Provoca primera conversión
   Outb (Reg_CLI, 0);   Outb (Reg_START, 0);
   loop
      if Nuevo_Dato then
         Nuevo_Dato := False;
         --  Procesa dato
         ...
      else
         -- hace otra cosa
         ...
      end if;
   end loop;
end Lee_AD_Con_Interrupciones;
```

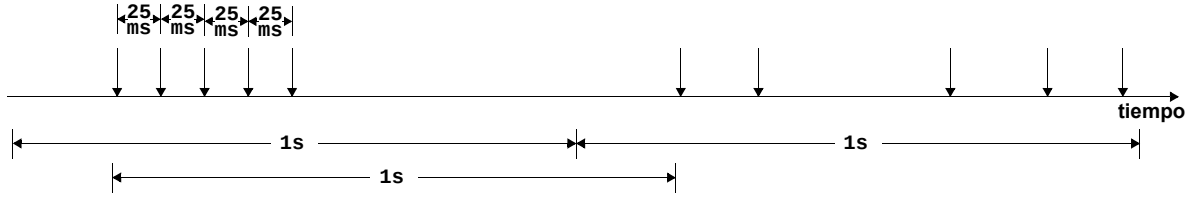

Técnicas básicas de entrada/salida: Buffer circular

UNIVERSIDAD DE CANTABRIA

En ocasiones es necesario disponer de un "buffer" de almacenamiento intermedio entre la puerta de E/S y la aplicación

Ejemplo: aplicación que procesa los datos recibidos por el puerto paralelo

- **mínimo intervalo de tiempo entre llegada de 2 datos: *25ms***
- **máximo número de datos en un segundo: *5***
- **en el procesado de cada dato se tarda *100ms* (10 datos/seg)**

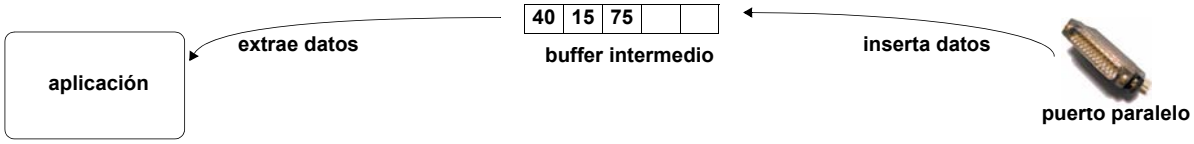


Técnicas básicas de entrada/salida: Buffer circular (cont.)

UNIVERSIDAD DE CANTABRIA

Si llegan los datos muy seguidos la aplicación no es capaz de procesarlos a tiempo

- **pero podemos almacenarlos temporalmente hasta que haya tiempo de realizar su procesado**
- **en este caso un buffer de longitud 5 (o 4) sería suficiente**

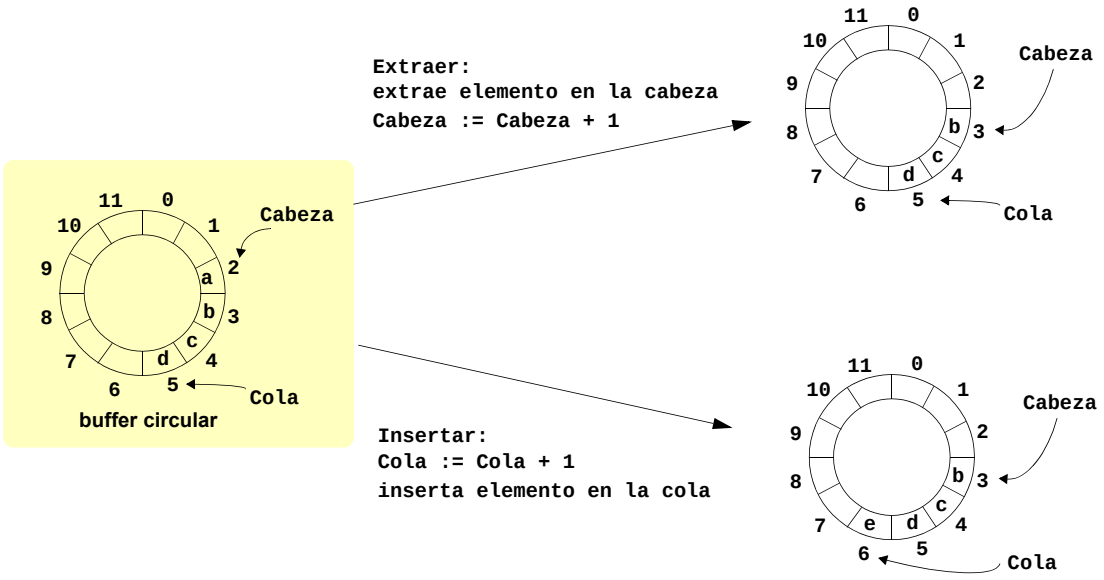


Cada vez que se extrae un elemento hay que desplazar los demás

- **esta operación puede consumir mucho tiempo**
- **para evitar este problema se utiliza un buffer circular**

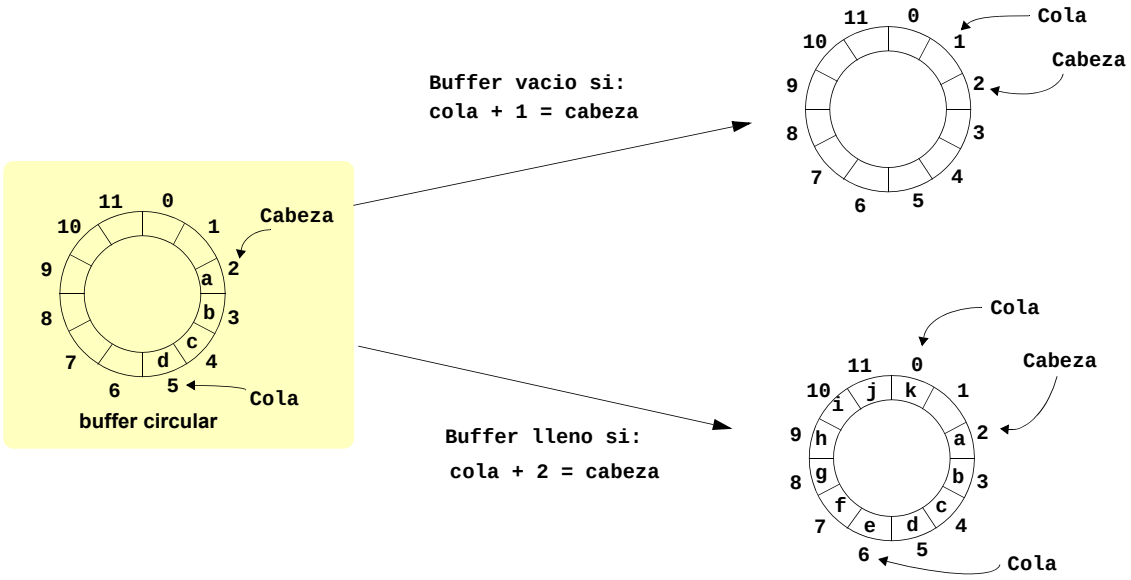
Buffer circular: Implementación

UNIVERSIDAD DE CANTABRIA



Buffer circular: Implementación (cont.)

UNIVERSIDAD DE CANTABRIA



Ejemplo de uso de un buffer circular

UNIVERSIDAD DE CANTABRIA

```
with MaRTE_OS;
with MaRTE_Hardware_Interrupts, IO_Interface;
use MaRTE_Hardware_Interrupts, IO_Interface;
with Basic_Integer_Types; use Basic_Integer_Types;
with System;
with Text_IO; use Text_IO;

procedure Buffer_Circular is

   --  Buffer circular
   type Indice_Buffer is mod 12; -- valores 0 .. 11
   type Array_Buffer is
     array (Indice_Buffer) of Unsigned_8;
   type Buffer_Circular is record
      Buf : Array_Buffer;
      Cabeza : Indice_Buffer := 0;
      Cola : Indice_Buffer := Indice_Buffer'Last - 1;
   end record;
```

Ejemplo de uso de un buffer circular (cont.)

UNIVERSIDAD DE CANTABRIA

```
function Buffer_Lleno (BC : Buffer_Circular)
                       return Boolean is
begin
   return BC.Cola + 2 = BC.Cabeza;
end Buffer_Lleno;


function Buffer_Vacio (BC : Buffer_Circular)
                       return Boolean is
begin
   return BC.Cola + 1 = BC.Cabeza;
end Buffer_Vacio;
```


Ejemplo de uso de un buffer circular (cont.)

UNIVERSIDAD DE CANTABRIA

```
procedure Buffer_Inserta
   (BC : in out Buffer_Circular;
    B  : in     Unsigned_8) is
begin
   if Buffer_Lleno (BC) then
      Put_Line ("Error: buffer lleno");
   else
      BC.Cola := BC.Cola + 1;
      BC.Buf (BC.Cola) := B;
   end if;
end Buffer_Inserta;
```

Ejemplo de uso de un buffer circular (cont.)

UNIVERSIDAD DE CANTABRIA

```
procedure Buffer_Extrae
   (BC : in out Buffer_Circular;
    B  : out    Unsigned_8) is
begin
   if Buffer_Vacio (BC) then
      Put_Line ("Error: buffer vacio");
   else
      B := BC.Buf (BC.Cabeza);
      BC.Cabeza := BC.Cabeza + 1;
   end if;
end Buffer_Extrae;

-- Registros del puerto paralelo
PP_BASE        : constant IO_Port := 16#378#;
PP_DATA_REG    : constant IO_Port := PP_BASE + 0;
PP_STATUS_REG  : constant IO_Port := PP_BASE + 1;
PP_CONTROL_REG : constant IO_Port := PP_BASE + 2;
```


Ejemplo de uso de un buffer circular (cont.)

UNIVERSIDAD DE CANTABRIA

```
--  Buffer compartido por el manejador y la aplicación
Buffer : Buffer_Circular;
pragma Volatile (Buffer);


--  Manejador de la interrupción del puerto paralelo
function Manejador_PP (Area : in System.Address;
                       Intr : in Hardware_Interrupt)
                      return Handler_Return_Code is
begin
   --  Lee dato y le almacena en el buffer circular
   Buffer_Inserta (Buffer, Inb (PP_DATA_REG));
   return POSIX_INTR_HANDLED_NOTIFY;
end Manejador_PP;
```

Ejemplo de uso de un buffer circular (cont.)

UNIVERSIDAD DE CANTABRIA

```
   Dato : Unsigned_8;

begin
   --  Instala manejador de interrupción
   if Associate (PARALLEL1_INTERRUPT,
                 Manejador_PP'Unrestricted_Access,
                 System.Null_Address, 0) /= 0 then
      Put_Line ("Error: Associate");
   end if;

   --  Configura como puerto de entrada y
   --  habilita interrupciones
   IO_Interface.Outb_P (PP_CONTROL_REG, 2#00_1_1_0000#);

   --  Habilita interrupción en el PIC
   if Unlock (PARALLEL1_INTERRUPT) /= 0 then
      Put_Line ("Error: Unlock");
   end if;
```


Ejemplo de uso de un buffer circular (cont.)

UNIVERSIDAD DE CANTABRIA

```
   loop
      while not Buffer_Vacio (Buffer) loop
         Buffer_Extrae (Buffer, Dato);
         -- procesa el dato leido, mientras dura el
         -- el procesado los bytes se irán almacenando
         -- en el buffer
         ...
      end loop;
      --  el programa puede ir haciendo otras cosas y,
      --  entre tanto, los bytes que van llegando se
      --  almacenan en el buffer
      ...
   end loop;

end Buffer_Circular;
```