Introducción

- A diferencia de otras arquitecturas, los puertos de E/S estan mapeados en la memoria de datos.
- En lugar de utilizar instrucciones especiales como *IN* y *OUT* como haciamos en el *MSX88*, aca usamos las instrucciones *LD* y *SD* para acceder a los puertos de entrada salida.
- Los puertos de E/S suelen estar posicionados en un lugar fuera del rango de la memoria de datos.
- En el WinMIPS64, la memoria de datos llega hasta la posición 0x0400. Los puertos de E/S empiezan en la posición 0x10000.

Puertos de Entrada / Salida

- Tenemos solamente dos puertos de E/S: *CONTROL* y *DATA*.
- *CONTROL* y *DATA* son direcciones de memoria fijas.
- Aplicando distintos comandos a través de *CONTROL*, es posible producir salidas o ingresar datos a través de la dirección *DATA*.
- Una vez cargado el registro *CONTROL*, se realiza el Servicio de Entrada/Salida.
  - Si necesito pasar algún parámetro, debo cargar *DATA* antes de invocar al servicio
  - Si el servicio devuelve un dato, debo leer *DATA* despues de invocar al servicio
- Las direcciones de memoria de *CONTROL* y *DATA* son 0x10000 y 0x10008 respectivamente.

Puertos de Entrada / Salida

Como el set de instrucciones del WinMIPS64 no cuenta con instrucciones que permitan cargar un valor inmediato de más de 16 bits, resulta conveniente definirlas en la memoria de datos, así:

```
.data
CONTROL: .word32 0x10000
DATA: .word32 0x10008
   ...
.code
   ...
lwu $s0, DATA($0)     ; 0x10000 en s0
lwu $s1, CONTROL($0) ; 0x10008 en s1
```

Servicios de Entrada / Salida

WinMIPS64 posee una terminal donde se pueden realizar las operaciones de Entrada/Salida

- Se puede utilizar como terminal alfanumerica donde:
  - Se puede imprimir valores y cadenas de caracteres
  - Leer números enteros, números de punto flotante o caracteres
- Se puede utilizar como una terminal gráfica:
  - Tiene una resolución de 50x50 puntos

Limpiar Pantalla

Para limpiar la pantalla hay que:

- Cargar en *CONTROL* el valor 6.

Impresión de un número

Para imprimir un número debo:

- Poner en *DATA* el valor a imprimir, ya se un entero o un flotante.
- Cargar en *CONTROL*:
  - Un *1* si el número es entero sin signo
  - Un *2* si el número es entero con signo
  - Un *3* si el número es de punto flotante

Impresión de una cadena

Para imprimir una cadena de caracteres:

- Asegurarme que la cadena finaliza con el carácter ASCII 0 (acordarse tipo de dato *.asciiz*)
- Poner en *DATA* la posición de memoria donde esta la cadena.
- Cargar en *CONTROL* el valor 4.

Leer un Número

Para la lectura de un número debo:

- Cargar en *CONTROL* el valor 8.
- Leer de *DATA* el valor númerico leido
  - Si el número ingresado tiene un punto(".") se lo convertira a punto flotante, sino será entero.
  - No hay manera fácil de saber si el usuario escribio un número entero o flotante desde el programa

Leer un Caracter

Para la lectura de un caracter debo:

- Cargar en *CONTROL* el valor 9.
- Leer de *DATA* el caracter
  - Se leerá el código ASCII del caracter leido
  - Como el caracter es de 8bits, nos conviene utilizar la instrucción *lbu* para su lectura.

Introducción

- La pantalla gráfica cuenta con una resolución de 50x50 puntos
- El punto (0, 0) es el punto en la esquina inferior izquierda
- El punto (49, 49) es el punto en la esquina superior derecha
- Cada punto puede tener un color independiente de los otros puntos
- Cada color esta compuesto de 3 partes:
  - R (Red): Rojo
  - G (Green): Verde
  - B (Blue): Azul
- Cada componente RGB, va de 0 a 255 (8 bits):
  - 0 : Ese color no esta presente
  - 255 : Fuerte presencia de ese color

Limpiar Pantalla

Para limpiar la pantalla hay que:

- Cargar en *CONTROL* el valor 7.

Dibujar un punto

Para pintar con un color indicado un punto de la pantalla gráfica:

- Escribir en *DATA* un color expresado en RGB (usando 4 bytes para representarlo)
  - un byte para cada componente de color e ignorando el valor del cuarto byte
- En *DATA*+4 la coordenada Y
- En *DATA*+5 la coordenada X
- En *CONTROL* un 5