

# Buses

## Organización del Computador I

David Alejandro González Márquez

Departamento de Computación  
Facultad de Ciencias Exactas y Naturales  
Universidad de Buenos Aires

01.03.2018

# Agenda

- Introducción
- Tipos de Bus
- Protocolos
- Ejercicios

# ¿Qué es un BUS?

# ¿Qué es un BUS?

## Bus

Es un canal encargado de transferir datos entre componentes o dispositivos dentro de una computadora. Se compone por un conjunto de *cables* o líneas y un *protocolo* de control.

# Tipos de Bus (algunos)

- Modo de sincronización
  - **Sincrónicos** (o síncronos): Si tienen una señal periódica (¡c!k!) que sirve para sincronizar los distintos dispositivos
  - **Asincrónicos** (o asíncronos): Si no la sincronización esta dada por eventos (conjuntos de cambios sobre las líneas)
- Uso de las líneas
  - **Dedicadas**: Si cada línea tiene un propósito específico
  - **Multiplexadas**: Si las líneas varían su propósito de acuerdo a la etapa del *protocolo*
- Modo de transferencia
  - **Serial**: La información se envía de forma secuencial bit tras bit
  - **Paralelo**: La información se envía en conjunto por todas las líneas

# Tipos de Bus (otros)

- Arbitraje

- **Centralizado:** Existe un dispositivo encargado de arbitrar el uso del bus
- **Descentralizado:** No existe un arbitro, los dispositivos acordan el uso en conjunto

- Uso del bus

- **Único maestro:** Solo un dispositivo controla el bus y transfiere datos de otros dispositivos
- **Múltiples maestros:** Múltiples dispositivos pueden controlar el bus

# Protocolos

# Protocolos

## ¿Qué es un protocolo?

Un *conjunto de reglas* que le indican a cada componente lo que *puede/debe* hacer en cada momento para que la comunicación sea *exitosa*

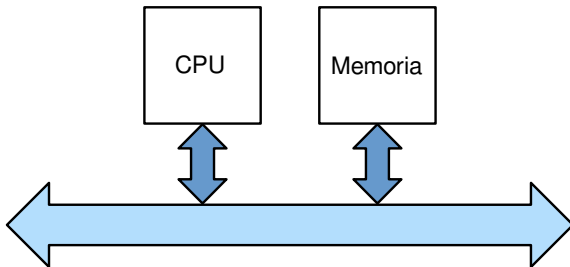
## Dos protocolos posibles

- **Transferencia:** Enviar información desde un dispositivo a otro
- **Adquisición:** Determinar cual es el master del bus, que dispositivo puede iniciar transferencias (árbitro)

Ojo, No confundir. El *clock* del bus con el *clock* de la CPU. La frecuencia del primero es mucho mas chica que la del segundo. Los dos se usan para sincronizar, pero de formas distintas.

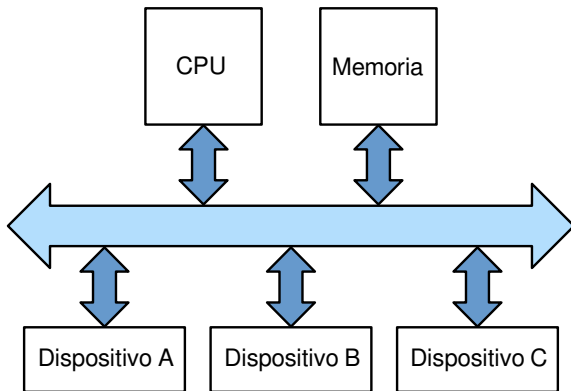


## En una Maquina Orga1



¿Qué debería viajar por el bus al ejecutar un `MOV [0x0014], R0`?

## En una Maquina Orga1i



¿Y si se ejecutara un `MOV R0, [0xFFF0]`?

# Propuesta de Bus para la Maquina Orga1

Líneas,

- `clk` : Señal de reloj
- `req` : Indicación de consulta
- `r/ $\overline{w}$`  : Señal de escritura
- `ack` : Confirmación de transmisión
- `data` : Conjunto de líneas de datos y direcciones

# Propuesta de Bus para la Maquina Orga1i

Líneas,

- $clk$  : Señal de reloj
- $req$  : Indicación de consulta
- $r/\overline{w}$  : Señal de escritura
- $ack$  : Confirmación de transmisión
- $data$  : Conjunto de líneas de datos y direcciones
- $mem/\overline{dev}$  : Indicación de memoria o dispositivo

# Diagrama de tiempo

## Ejercicio 1

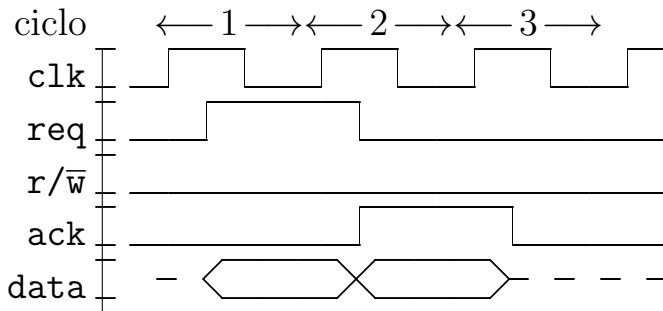
Utilizando la propuesta anterior, hacer el diagrama de tiempos del bus de una máquina Orga1 para una escritura en memoria.

Teniendo en cuenta que:

- cada agente (CPU, memoria, dispositivo) puede cargar o censar las líneas
- sólo se cargan las líneas cuando el *clock* está alto
- sólo se censan las líneas cuando el *clock* está bajo
- al finalizar el ciclo el bus debe volver al estado inicial

# Diagrama de tiempo - Solución Ejercicio 1

Escritura en memoria para la Orga1



# Diagrama de tiempo

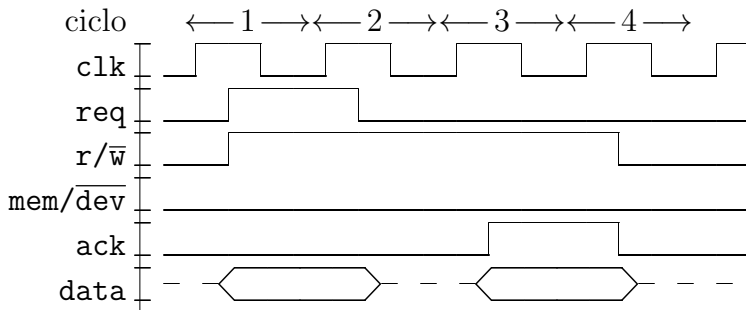
## Ejercicio 2

Hacer el diagrama de tiempos para una lectura desde un dispositivo en Orga1i. Los dispositivos requieren de al menos un ciclo para responder una lectura.

# Diagrama de tiempo

## Ejercicio 2

Hacer el diagrama de tiempos para una lectura desde un dispositivo en Orga1i. Los dispositivos requieren de al menos un ciclo para responder una lectura.





# Capacidad del bus

## Ejercicio 3

Suponiendo que el clock tiene una frecuencia de 100 Khz

- 1 ¿Cuánto tiempo toma realizar un ciclo de escritura a memoria? (ejercicio 1)
- 2 ¿Cuánto toma un ciclo de lectura de un dispositivo? (ejercicio 2)
- 3 Si cada operación es de 16 bits, ¿cuál es la máxima capacidad del bus?

## Capacidad del bus - Solución Ejercicio 3

*Pista*  $\rightarrow 100 \text{ KHz} = 100.000 \text{ ciclos por segundo.}$

1 Escritura = 3 ciclos =  $3 \times 10^{-5} \approx \frac{1}{33333}$  segundos.

2 Lectura = 4 ciclos =  $4 \times 10^{-5} = \frac{1}{25000}$  segundos.

3 Máxima capacidad del bus para escritura = 33333 operaciones por segundo = 66666 bytes por segundo  $\approx 65.10 \text{ KB/s.}$

Máxima capacidad del bus para lectura = 25000 operaciones por segundo = 50000 bytes por segundo  $\approx 48.82 \text{ KB/s.}$

# Diagrama de tiempo

## Ejercicio 4

Sea un bus síncrono entre un CPU y una memoria cuyo protocolo de escritura es:

Señales: REQ,  $\overline{RD}/W$ , dir, data, DWAIT y ACK

1. La CPU:

- coloca la dirección que desea leer en el bus de direcciones
- sube la línea de  $\overline{RD}/W$  para indicar que es una escritura
- sube la línea de REQ para indicar que quiere un acceso a memoria

2. El módulo de memoria:

- detecta la señal alta en REQ
- toma la dirección del bus de direcciones
- sube la señal ACK para indicar al CPU que responderá al pedido y
- sube DWAIT para indicar que está esperando el dato

3. La CPU coloca el dato en el bus de datos y baja la línea de  $\overline{RD}/W$

4. El módulo de memoria toma el dato del bus de datos y baja la señal DWAIT

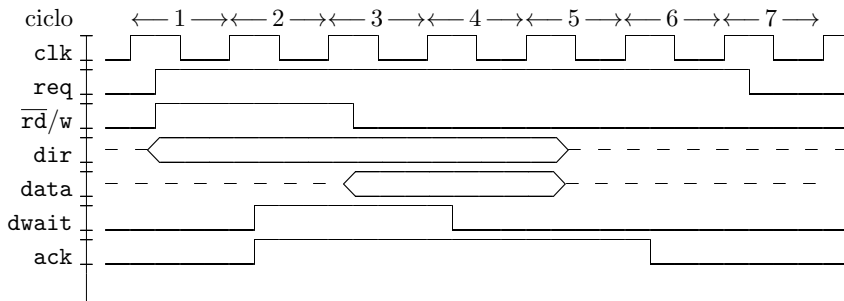
5. La CPU detecta la señal baja en DWAIT y quita la dirección y el dato

6. La memoria indica la finalización de la operación bajando ACK

7. La CPU detecta esto y baja REQ

Dibujar el diagrama de tiempos del protocolo.

# Solución Ejercicio 4



# Arbitraje

## Ejercicio 5

Un bus síncrono comunica al procesador con hasta 3 dispositivos adicionales.

A cada uno se le asigna una prioridad única y una línea exclusiva de solicitud de transferencia  $S_i$ . El bus tiene 3 de estas líneas, siendo  $S_1$  la de mayor prioridad.

Si durante el ciclo  $T_k$  un dispositivo necesita el bus, debe esperar hasta el inicio del ciclo  $T_{k+1}$ , y conjuntamente con el flanco de reloj, levantar su línea de solicitud.

Cuando está por finalizar el ciclo  $T_{k+1}$ , todos los dispositivos con petición pendiente deben verificar el estado de las otras dos líneas  $S_i$ ; el dispositivo que obtuvo el bus debe bajar su línea de solicitud, mientras que los restantes dispositivos con petición pendiente deben mantener las suyas.

- 1 Considerar un nuevo dispositivo de menor prioridad (4). ¿Necesita línea de solicitud. ¿Por qué?
- 2 Este esquema de arbitraje distribuido hace que uno de los dispositivos tenga un tiempo de espera de bus inferior a los restantes cuando el bus está disponible. Decir cuál es este dispositivo y explicar claramente por qué.

## Arbitraje - Solución Ejercicio 5

- 1 El dispositivo de menor prioridad utilizará el bus en cualquier ciclo libre donde todas las señales de solicitud esten en bajo.
- 2 El dispositivo de mayor prioridad es el que debe esperar menos para obtener el bus, ya que su solicitud es mandatoria sobre el resto de los dispositivos.

# ¿Preguntas?



# ¿Preguntas?



He-Man