Clase 16 - Entradas y Salidas: Comunicación

IIC2343 - Arquitectura de Computadores

Profesor:
- Felipe Valenzuela González
Correo:
frvalenzuela@alumni.uc.cl

Resumen de clase pasada

Arquitectura de Computadores: Dispositivos I/O

- Todos los dispositivos que no sean CPU o memoria, y se comuniqu con ellos, son llamados dispositivos de I/O

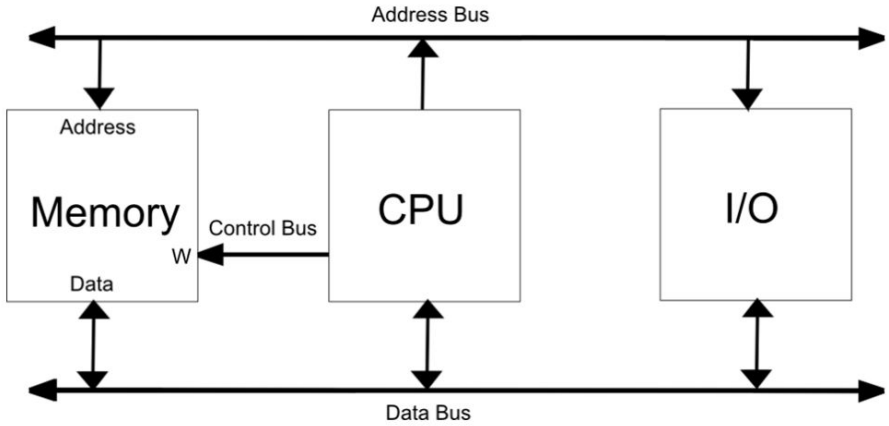


Imagen Referencia: https://github.com/IIC2343/Syllabus-anteriores/blob/main/Apuntes%20(viejos)/8%20-%20Comunicaci%C3%B3n%20de%20CPU%20y%20Memoria%20con%20IO.pdf

Arquitectura de Computadores: Dispositivos I/O

- Un dispositivo de I/O tiene un controlador encargado de comunicarse con la CPU y la memoria, y de controlar la parte electromecánica. Todos los dispositivos que no sean CPU o memoria, y se comuniquen con ellos, son llamados dispositivos de I/O

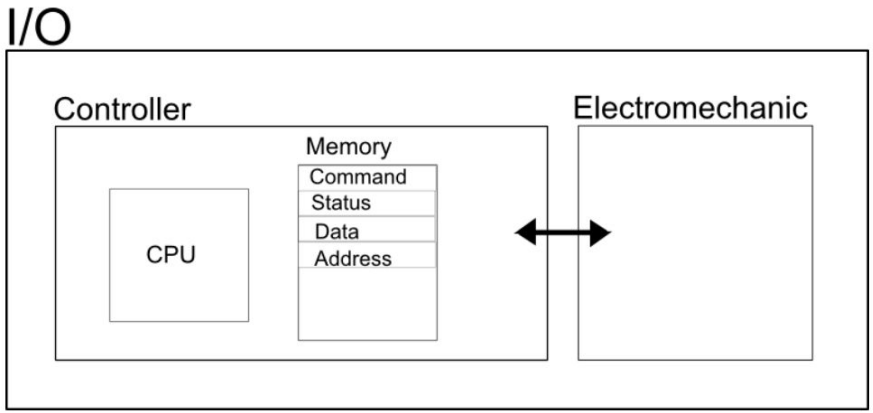


Imagen Referencia: https://github.com/IIC2343/Syllabus-anteriores/blob/main/Apuntes%20(viejos)/8%20-%20Comunicaci%C3%B3n%20de%20CPU%20y%20Memoria%20con%20IO.pdf

Arquitectura de Computadores: Memory Map

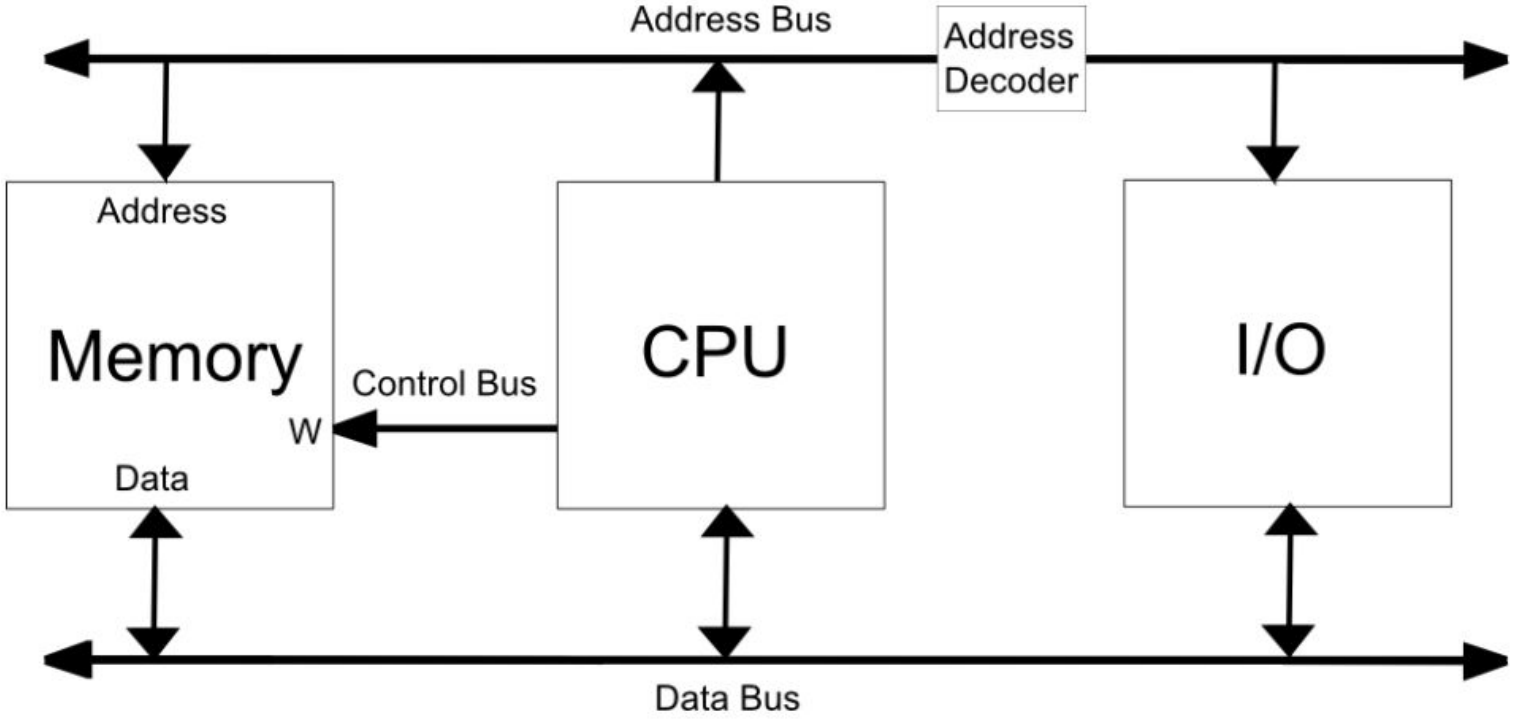


Imagen Referencia: https://github.com/IIC2343/Syllabus-anteriores/blob/main/Apuntes%20(viejos)/8%20-%20Comunicaci%C3%B3n%20de%20CPU%20y%20Memoria%20con%20IO.pdf

Arquitectura de Computadores: Port I/O

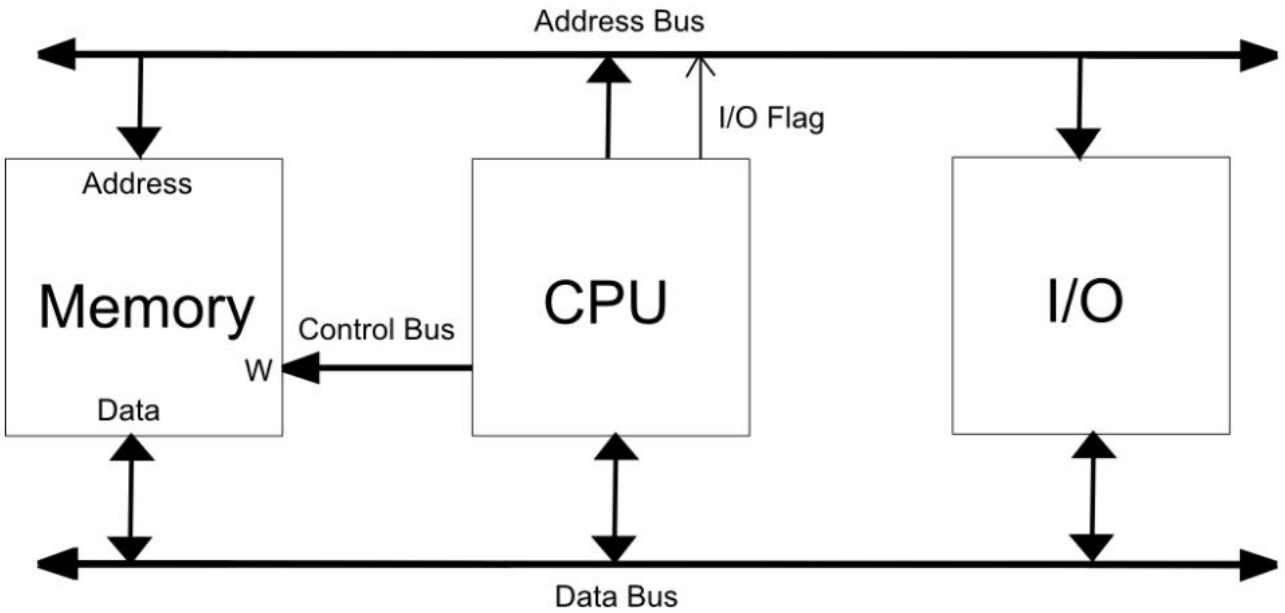


Imagen Referencia: https://github.com/IIC2343/Syllabus-anteriores/blob/main/Apuntes%20(viejos)/8%20-%20Comunicaci%C3%B3n%20de%20CPU%20y%20Memoria%20con%20IO.pdf

¿Dudas?

Arquitectura de Computadores: Dispositivos I/O

Usaremos como analogía a un curso haciendo una guía de ejercicios:

- Alumnos (I/O) hacen guía de ejercicios, si alguien termina, el profesor quiere guardar la respuesta.
- Profesor (CPU) está corrigiendo pruebas.
- Pizarra o proyector (Memoria), donde se pueden ver los ejercicios y anotar las respuestas.
- Comandos: instrucciones de parte del profesor.
- Estado: alumnos trabajando, con dudas.
- Datos: preguntas y respuestas guía, dudas alumnos, respuestas profesor.
- Supuestos: alumnos no hablan entre ellos, se entregan varias guías durante la clase

Imagen Referencia: https://github.com/IIC2343/Syllabus-anteriores/blob/main/Apuntes%20(viejos)/8%20-%20Comunicaci%C3%B3n%20de%20CPU%20y%20Memoria%20con%20IO.pdf

Revisaremos tres modelos de interacción entre alumnos y profesor

Sin proyector ni pizarrón ni copias de las guías, alumnos tímidos:

1. **Polling**

Imagen Referencia: https://github.com/IIC2343/Syllabus-anteriores/blob/main/Apuntes%20(viejos)/8%20-%20Comunicaci%C3%B3n%20de%20CPU%20y%20Memoria%20con%20IO.pdf

Arquitectura de Computadores: Polling

- ¿Necesitamos cambio de hardware?

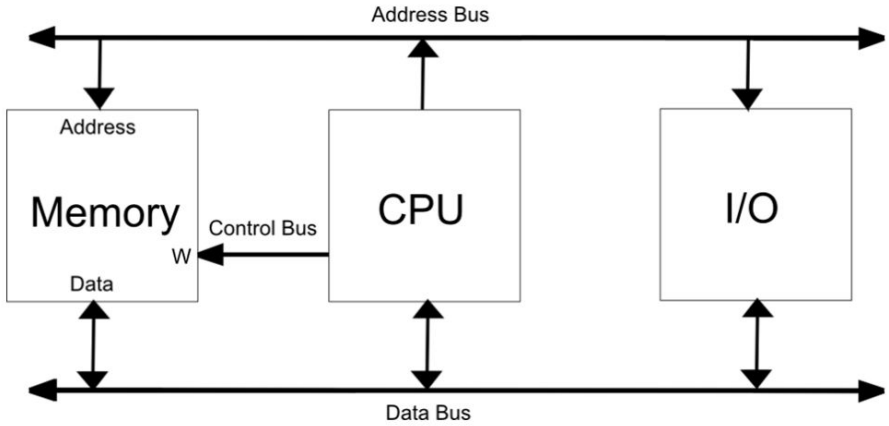


Imagen Referencia: https://github.com/IIC2343/Syllabus-anteriores/blob/main/Apuntes%20(viejos)/8%20-%20Comunicaci%C3%B3n%20de%20CPU%20y%20Memoria%20con%20IO.pdf

Arquitectura de Computadores: Polling

- ¿Necesitamos cambio de hardware?
- **NO**

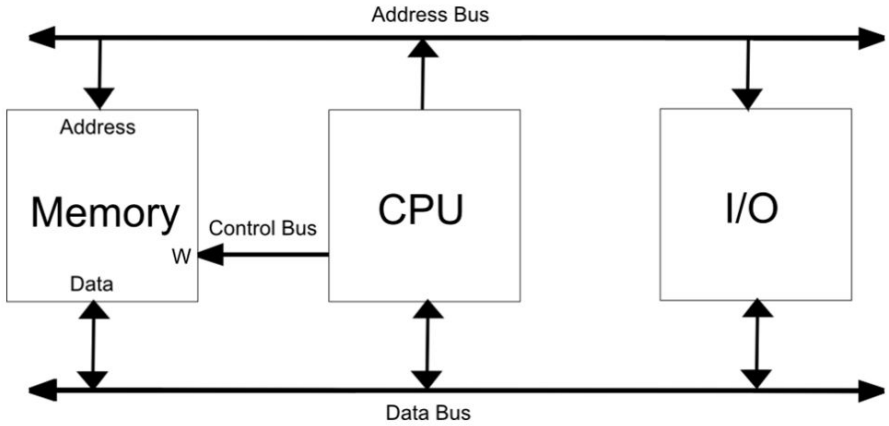


Imagen Referencia: https://github.com/IIC2343/Syllabus-anteriores/blob/main/Apuntes%20(viejos)/8%20-%20Comunicaci%C3%B3n%20de%20CPU%20y%20Memoria%20con%20IO.pdf

Revisaremos tres modelos de interacción entre alumnos y profesor

Sin proyector ni pizarrón ni copias de las guías, alumnos tímidos:

1. **Polling**

Sin proyector ni pizarrón ni copias de las guías, alumnos

preguntones y participativos:

2. **Interrupción**

Imagen Referencia: https://github.com/IIC2343/Syllabus-anteriores/blob/main/Apuntes%20(viejos)/8%20-%20Comunicaci%C3%B3n%20de%20CPU%20y%20Memoria%20con%20IO.pdf

Arquitectura de Computadores: Interrupción

- ¿Necesitamos cambio de hardware?

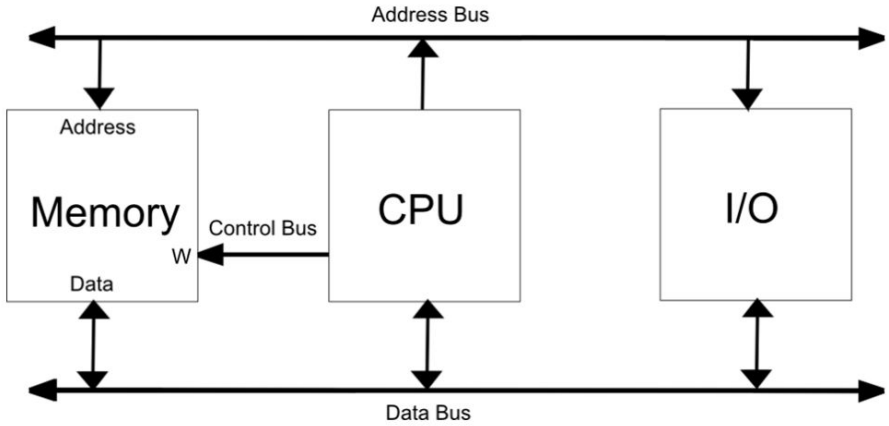


Imagen Referencia: https://github.com/IIC2343/Syllabus-anteriores/blob/main/Apuntes%20(viejos)/8%20-%20Comunicaci%C3%B3n%20de%20CPU%20y%20Memoria%20con%20IO.pdf

Arquitectura de Computadores: Interrupción

- ¿Necesitamos cambio de hardware?
- SI, una señal de **solicitud de interrupción**

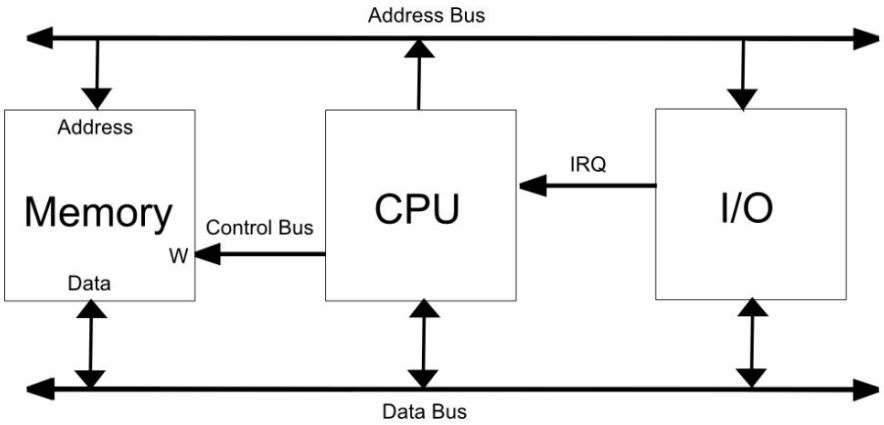


Imagen Referencia: https://github.com/IIC2343/Syllabus-anteriores/blob/main/Apuntes%20(viejos)/8%20-%20Comunicaci%C3%B3n%20de%20CPU%20y%20Memoria%20con%20IO.pdf

Arquitectura de Computadores: Interrupción

1. Dispositivo solicita interrumpir, enviando señal IRQ

2. CPU termina de ejecutar la instrucción actual y guarda condition

codes en el stack

3. CPU revisa si el flag de interrupciones está activo (IF = 1). En caso contrario, saltar al paso 11.

4. CPU deshabilita atención de más interrupciones (IF=0)

5. CPU llama a la ISR asociada al dispositivo

Arquitectura de Computadores: Interrupción

6. ISR respalda estado actual de la CPU

7. ISR es ejecutada

8. ISR restaura estado de la CPU

9. ISR retorna

10. CPU habilita la atención de interrupciones (IF=1)

11. CPU recupera condition codes desde el stack

Arquitectura de Computadores: Interrupción

- Al usar varios dispositivos, es necesario incorporar un controlador de interrupciones y una tabla de vectores de interrupción

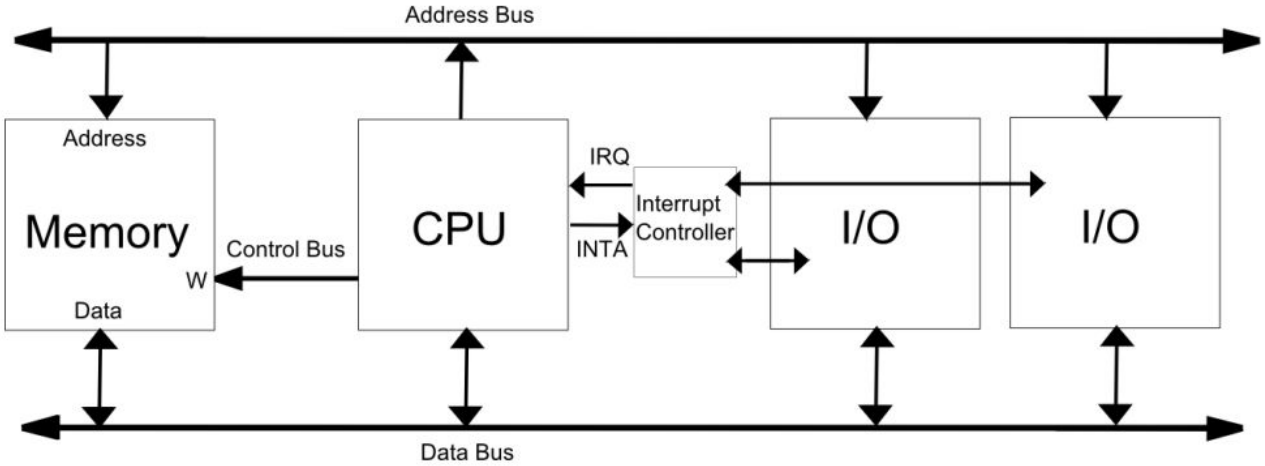


Imagen Referencia: https://github.com/IIC2343/Syllabus-anteriores/blob/main/Apuntes%20(viejos)/8%20-%20Comunicaci%C3%B3n%20de%20CPU%20y%20Memoria%20con%20IO.pdf

Arquitectura de Computadores: Interrupción

- En general, un controlador de interrupciones tendrá al menos los siguientes componentes:
- Registro de comandos y estado
- ***Interrupt Request Register:*** Registro de interrupciones en espera de atención
- ***In-Service Register:*** Registro de interrupciones en atención
- ***Interrupt Mask Register:*** Registro de enmascaramiento de interrupciones
- Circuito para manejar prioridades de interrupciones

Imagen Referencia: https://github.com/IIC2343/Syllabus-anteriores/blob/main/Apuntes%20(viejos)/8%20-%20Comunicaci%C3%B3n%20de%20CPU%20y%20Memoria%20con%20IO.pdf

Tipos de interrupción en la CPU

- **Excepciones:** Condiciones de error al ejecutar una instrucción. Estas son atendidas por la CPU (en particular, el sistema operativo) a través de **ISRs** específicas: *exception handler*

  Ejemplos: división por cero, stack overflow.

- **Interrupciones de software (*traps*):** Son “llamadas al sistema operativo”, interrupciones explícitas en los programas para ceder el control al sistema operativo para que realice una acción

  Ejemplo: ecall

Arquitectura de Computadores: Interrupción Ejemplo

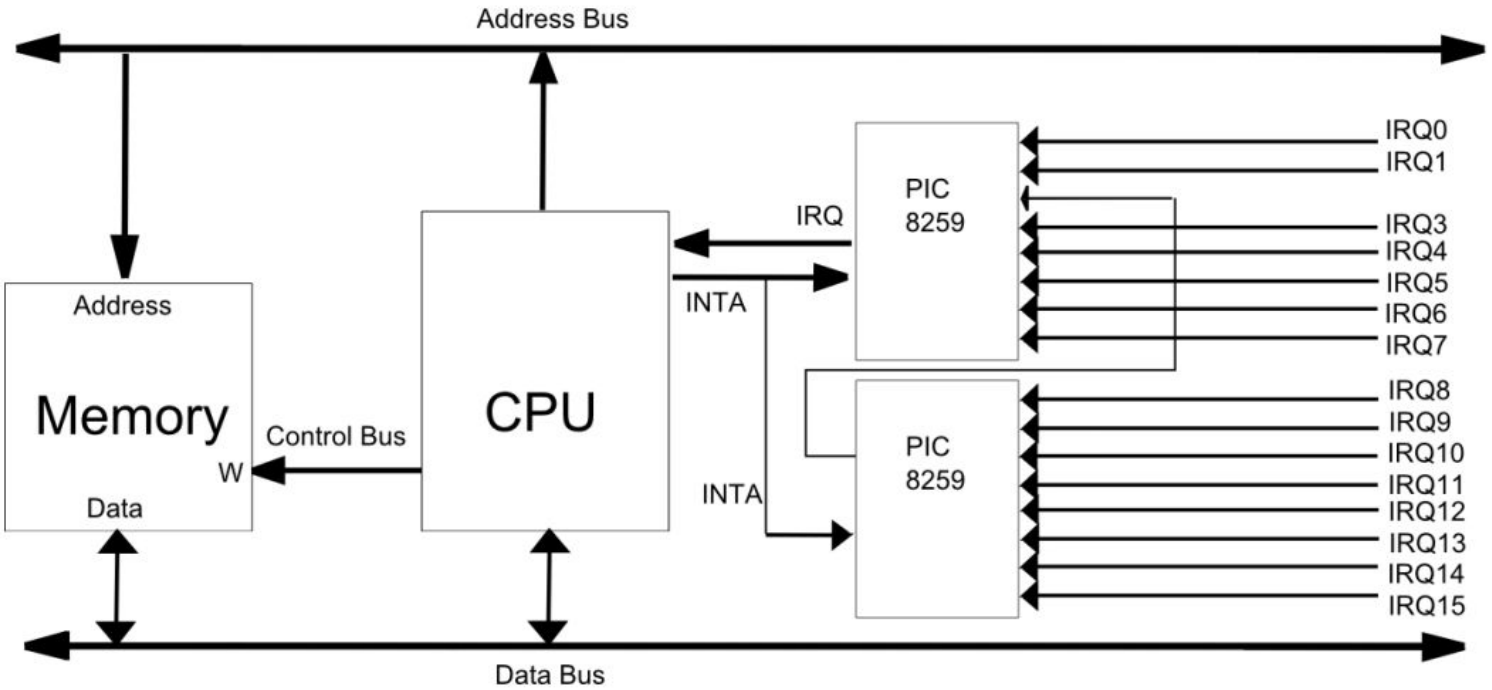


Imagen Referencia: https://github.com/IIC2343/Syllabus-anteriores/blob/main/Apuntes%20(viejos)/8%20-%20Comunicaci%C3%B3n%20de%20CPU%20y%20Memoria%20con%20IO.pdf

Arquitectura de Computadores: Interrupción Ejemplo

IRQ	Dispositivo	Vector de interrupción
IRQ0	Timer del sistema	08
IRQ1	Puerto PS/2: Teclado	09
IRQ2	Conectada al PIC secundario	0A
IRQ3	Puerto serial	0B
IRQ4	Puerto serial	0C
IRQ5	Puerto paralelo	0D
IRQ6	Floppy disk	0E
IRQ7	Puerto paralelo	0F
IRQ8	Real time clock (RTC)	70
IRQ9-11	No tienen asociación estándar, libre uso.	71-73
IRQ12	Puerto PS/2: Mouse	74
IRQ13	Coprocesador matemático	75
IRQ14	Controlador de disco 1	76
IRQ15	Controlador de disco 2	77

Imagen Referencia: https://github.com/IIC2343/Syllabus-anteriores/blob/main/Apuntes%20(viejos)/8%20-%20Comunicaci%C3%B3n%20de%20CPU%20y%20Memoria%20con%20IO.pdf

Arquitectura de Computadores: Interrupción Ejemplo

Dirección del vector	Tipo	Función asociada
00-01	Excepción	Exception handlers
02	Excepción	Usada para errores críticos del sistema, no enmascarable
03-07	Excepción	Exception handlers
08	IRQ0	Timer del sistema
09	IRQ1	Puerto PS/2: Teclado
0A	IRQ2	Conectada al PIC secundario
0B	IRQ3	Puerto serial
0C	IRQ4	Puerto serial
0D	IRQ5	Puerto paralelo
0E	IRQ6	Floppy disk
0F	IRQ7	Puerto paralelo
10	Int. de Software	Funciones de video
11-6F	Int. de Software	Funciones varias
70	IRQ8	Real time clock (RTC)
71 - 73	IRQ9-11	No tienen asociación estándar, libre uso
74	IRQ12	Puerto PS/2: Mouse
75	IRQ13	Coprocesador matemático
76	IRQ14	Controlador de disco 1
77	IRQ15	Controlador de disco 2

Imagen Referencia: https://github.com/IIC2343/Syllabus-anteriores/blob/main/Apuntes%20(viejos)/8%20-%20Comunicaci%C3%B3n%20de%20CPU%20y%20Memoria%20con%20IO.pdf

¿Dudas?

Revisaremos tres modelos de interacción entre alumnos y profesor

Sin proyector ni pizarrón ni copias de las guías, alumnos tímidos:

1. **Polling**

Sin proyector ni pizarrón ni copias de las guías, alumnos

preguntones y participativos:

2. **Interrupción**

Pizarra interactiva, alumnos preguntones:

3. **Interrupción con acceso directo a memoria**

Imagen Referencia: https://github.com/IIC2343/Syllabus-anteriores/blob/main/Apuntes%20(viejos)/8%20-%20Comunicaci%C3%B3n%20de%20CPU%20y%20Memoria%20con%20IO.pdf

Arquitectura de Computadores: DMA

- Para permitir que los I/O tengan acceso directo a memoria se utiliza un controlador DMA

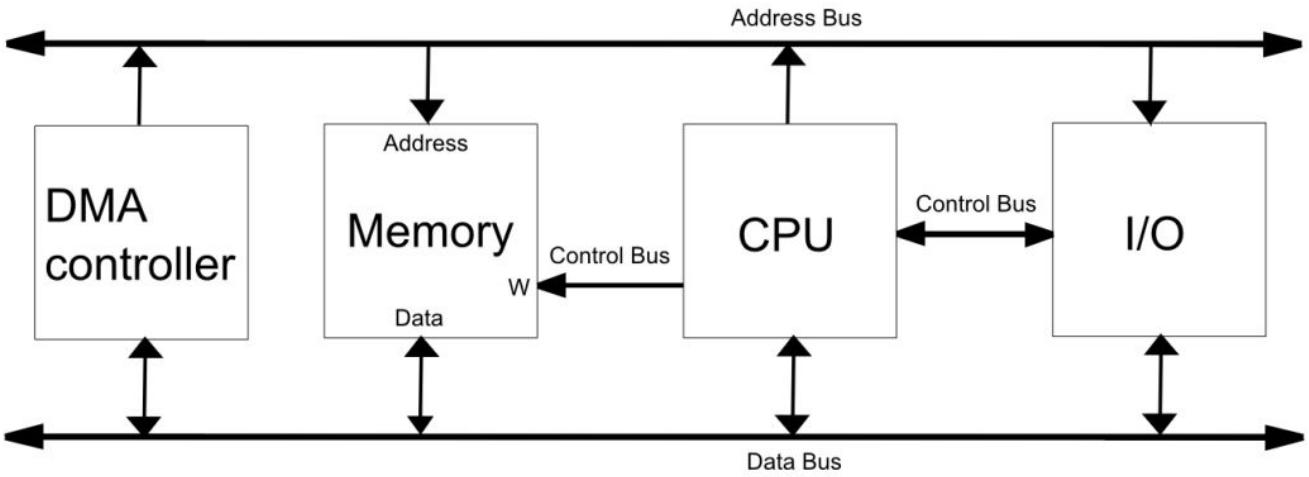


Imagen Referencia: https://github.com/IIC2343/Syllabus-anteriores/blob/main/Apuntes%20(viejos)/8%20-%20Comunicaci%C3%B3n%20de%20CPU%20y%20Memoria%20con%20IO.pdf

Arquitectura de Computadores: DMA

- En general, un controlador de DMA tendrá al menos los siguientes componentes:
  - Registro de comandos y estado
  - Registros de dirección de origen y destino
  - Registro contador de palabras
  - Buffer de almacenamiento temporal
  - Unidad de control

Imagen Referencia: https://github.com/IIC2343/Syllabus-anteriores/blob/main/Apuntes%20(viejos)/8%20-%20Comunicaci%C3%B3n%20de%20CPU%20y%20Memoria%20con%20IO.pdf

Arquitectura de Computadores: DMA e Interrupciones

- Arquitectura de un computador con memory mapped I/O, interrupciones y DMA

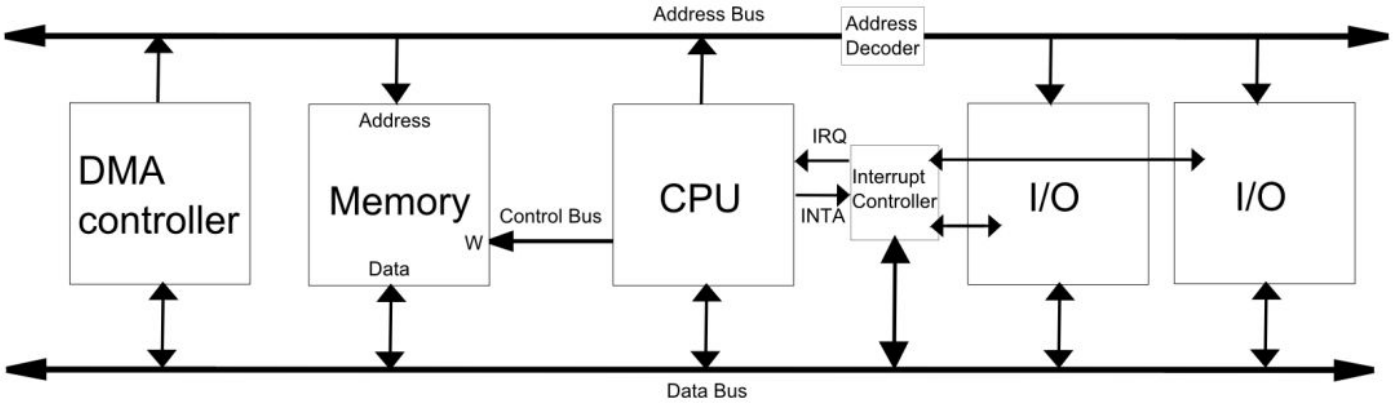


Imagen Referencia: https://github.com/IIC2343/Syllabus-anteriores/blob/main/Apuntes%20(viejos)/8%20-%20Comunicaci%C3%B3n%20de%20CPU%20y%20Memoria%20con%20IO.pdf

¿Dudas?

Ejercicio I2-2025-1

(a) (1.5 ptos.) En el pasado, la comunicación con dispositivos I/O se diseñó mediante mapeo a puertos como alternativa al mapeo a memoria, debido a una desventaja que este último presentaba en ese entonces. Hoy en día, esa limitación ha dejado de ser relevante, por lo que el mapeo a memoria es el que predomina en arquitecturas modernas. Explique cuál era la desventaja que tenía el mapeo a memoria en ese contexto, y por qué en la actualidad ya no representa un problema.

Ejercicio I2-2025-1

(a) (1.5 ptos.) En el pasado, la comunicación con dispositivos I/O se diseñó mediante mapeo a puertos como alternativa al mapeo a memoria, debido a una desventaja que este último presentaba en ese entonces. Hoy en día, esa limitación ha dejado de ser relevante, por lo que el mapeo a memoria es el que predomina en arquitecturas modernas. Explique cuál era la desventaja que tenía el mapeo a memoria en ese contexto, y por qué en la actualidad ya no representa un problema.

Solución: La desventaja que tenía el mapeo a memoria consistía en la *memory barrier*, *i.e.* la limitación del espacio direccionable por la reserva significativa de direcciones. Como en el pasado las memorias RAM tenían un tamaño reducido, esta pérdida era significativa e implicaba la imposibilidad de usar variables normalmente por la baja cantidad de direcciones disponibles. Hoy en día, en cambio, las memorias RAM poseen un tamaño lo suficientemente grande para que este problema sea inexistente. Se otorgan **0.75 ptos.** por responder correctamente, y **0.75 ptos.** por justificación.

Ejercicio I2-2025-1

(a) (1.5 ptos.) En el pasado, la comunicación con dispositivos I/O se diseñó mediante mapeo a puertos como alternativa al mapeo a memoria, debido a una desventaja que este último presentaba en ese entonces. Hoy en día, esa limitación ha dejado de ser relevante, por lo que el mapeo a memoria es el que predomina en arquitecturas modernas. Explique cuál era la desventaja que tenía el mapeo a memoria en ese contexto, y por qué en la actualidad ya no representa un problema.

Ejercicio I2-2025-1

(1.5 ptos.) Explique el uso e importancia de la señal IRQ enviada por el controlador de interrupciones a la CPU.

Ejercicio I2-2025-1

(1.5 ptos.) Explique el uso e importancia de la señal IRQ enviada por el controlador de interrupciones a la CPU.

Solución: La señal IRQ (*Interrupt Request*) es utilizada para comunicarle a la CPU que existe, al menos, un dispositivo I/O que requiere su atención. El controlador de interrupciones se encarga de su envío en caso de uno de los dispositivos de la arquitectura haya gatillado una interrupción. Es importante ya que es la que permite que inicie el proceso de interrupción, que termina en la ejecución de la ISR (*Interrupt Service Routine*) del dispositivo. Se otorgan **0.75 ptos.** por responder correctamente, y **0.75 ptos.** por justificación.

Ejercicio I2-2025-1

(1.5 ptos.) Entregue dos ejemplos de interrupciones gatilladas por la CPU y no por dispositivos I/O, indicando su tipo (excepción o *trap*) y explicando en qué consisten. Ambos ejemplos pueden ser de un mismo tipo.

Solución: A continuación, se listan dos ejemplos:

- ***Stack overflow*:** Es una **excepción** que se gatilla cuando el tamaño de la memoria *stack* supera el límite permitido, generando posibles errores de sobreescritura en variables.
- ***Environment call*:** Es una ***trap*** que permite cederle el control al sistema operativo para que realice una acción, como imprimir en consola o terminar un proceso.

Se otorgan **0.75 ptos.** por cada ejemplo correctamente descrito. Se otorga la mitad del puntaje en un ejemplo si su descripción no es correcta, o bien su tipo no es consistente con el descrito.

¿Dudas?

Ejercicio I2-2025-1

Aburrido/a de los quehaceres del hogar, decide automatizar el lavado y secado de su ropa a través de un sistema centralizado que hace uso de la ISA RISC-V y de tres dispositivos: una lavadora (`washer`), una secadora (`dryer`) y una catapulta (`catapult`). A continuación, una ilustración de su innovadora idea:

Ejercicio I2-2025-1

Para llevar a cabo la comunicación entre la CPU y los dispositivos, los registros de 32 bits de estos últimos son mapeados en memoria desde la dirección 0xFEE1DEAD:

Offset	Nombre	Descripción
0x00	`washer_status`	Registro de estado de la lavadora
0x04	`washer_command`	Registro de comando de la lavadora
0x08	`washer_water_level`	Configuración de nivel de agua de la lavadora
0x0C	`dryer_status`	Registro de estado de la secadora
0x10	`dryer_command`	Registro de comando de la secadora
0x14	`dryer_type`	Configuración de tipo de ropa a secar
0x18	`catapult_status`	Registro de estado de la catapulta
0x1C	`catapult_command`	Registro de comando de la catapulta

Ejercicio I2-2025-1

Los estados, comandos y configuraciones, por otra parte, se rigen a partir de las siguientes tablas:

Nombre	Descripción	Valor
`washer_status`	Apagada	0
`washer_status`	Encendida, puerta abierta	1
`washer_status`	Encendida, puerta cerrada	2
`washer_status`	Lavando	4
`washer_command`	Encender	1
`washer_command`	Apagar	255
`washer_command`	Abrir puerta	16
`washer_command`	Cerrar puerta	32
`washer_command`	Expulsar ropa	64
`washer_command`	Lavar	128

Nombre	Descripción	Valor
`dryer_status`	Apagada	0
`dryer_status`	Encendida, puerta abierta	1
`dryer_status`	Encendida, puerta cerrada	2
`dryer_status`	Secando	4
`dryer_command`	Encender	1
`dryer_command`	Apagar	255
`dryer_command`	Abrir puerta	16
`dryer_command`	Cerrar puerta	32
`dryer_command`	Expulsar ropa	64
`dryer_command`	Secar	128

Nombre	Descripción	Valor
`catapult_status`	Preparada	11
`catapult_status`	Disparada	13
`catapult_command`	Lanzar	11
`catapult_command`	Reposicionar	13
`washer_water_level`	Alto	1
`washer_water_level`	Medio	2
`washer_water_level`	Bajo	3
`dryer_type`	Ropa normal	1
`dryer_type`	Sábanas	2
`dryer_type`	Toallas	3

Para llevar a cabo la automatización, desarrolla el siguiente programa que asume que la ropa ya se encuentra dentro de la lavadora y que el lanzamiento de la catapulta es instantáneo:

Ejercicio I2-2025-1

```
washing_automation:
# Carga 0 + 0xFEE1DEAD en t0
addi t0, zero, 0xFEE1DEAD
addi s0, zero, 0
addi s1, zero, 1
addi s2, zero, 2
addi s3, zero, 3
addi s4, zero, 4
addi s5, zero, 16
addi s6, zero, 32
addi s7, zero, 64
addi s8, zero, 128
addi s9, zero, 255
addi s10, zero, 11
addi s11, zero, 13

step_one:
    lw t1, 0(t0) # Carga Mem[t0 + 0] en t1
    # Salta a continue_step_one si t1 >= s1
    bge t1, s1, continue_step_one
    sw s1, 4(t0) # Carga s1 en Mem[t0 + 4]
    continue_step_one:
      # Salta a step_two si t1 == s2
      beq t1, s2, step_two
      sw s6, 4(t0)

step_two:
    lw t1, 24(t0)
    beq t1, s10, step_three
    sw s11, 28(t0)

# Continúa en step_three -->
```

```
step_three:
    lw t1, 12(t0)
    bge t1, s1, continue_step_three
    sw s1, 16(t0)
    continue_step_three:
      beq t1, s1, step_four
      sw s5, 16(t0)

step_four:
    sw s1, 8(t0)
    sw s8, 4(t0)
    while_step_four:
        lw t1, 0(t0)
        beq t1, s4, while_step_four
    sw s5, 4(t0)
    sw s7, 4(t0)
    sw s9, 4(t0)

step_five:
    sw s10, 28(t0)
    sw s11, 28(t0)

step_six:
    sw s6, 16(t0)
    sw s3, 20(t0)
    sw s8, 16(t0)
    while_step_six:
        lw t1, 12(t0)
        beq t1, s4, while_step_six
    sw s5, 16(t0)
    sw s7, 16(t0)
    sw s9, 16(t0)
```

Ejercicio I2-2025-1

Describa, basándose en las tablas provistas, las tareas que realiza el programa en cada *label* `step_*`. Haga una descripción según los estados, comandos y configuraciones de los dispositivos; no una explicación textual de cada instrucción. Sea preciso/a respecto a las condiciones evaluadas en los saltos e indique, cuando corresponda, el momento en el que se continúa al siguiente `step`.

Ejercicio I2-2025-1

Describa, basándose en las tablas provistas, las tareas que realiza el programa en cada *label* `step_*`. Haga una descripción según los estados, comandos y configuraciones de los dispositivos; no una explicación textual de cada instrucción. Sea preciso/a respecto a las condiciones evaluadas en los saltos e indique, cuando corresponda, el momento en el que se continúa al siguiente `step`.

- `step_one` **(1 pto.)**: Se revisa el estado de la lavadora. Si está apagada, se prende. Luego, si está prendida con la puerta abierta, se cierra y salta a `step_two`. En otro caso, continúa a `step_two` directamente.
- `step_two` **(0.5 ptos.)**: Se revisa el estado de la catapulta. Si está preparada, se salta a `step_three`. En otro caso, se reposiciona y continúa a `step_three`.
- `step_three` **(1 pto.)**: Se revisa el estado de la secadora. Si está apagada, se prende. Luego, si está prendida con la puerta cerrada, se abre y salta a `step_four`. En otro caso, continúa a `step_four` directamente.

Ejercicio I2-2025-1

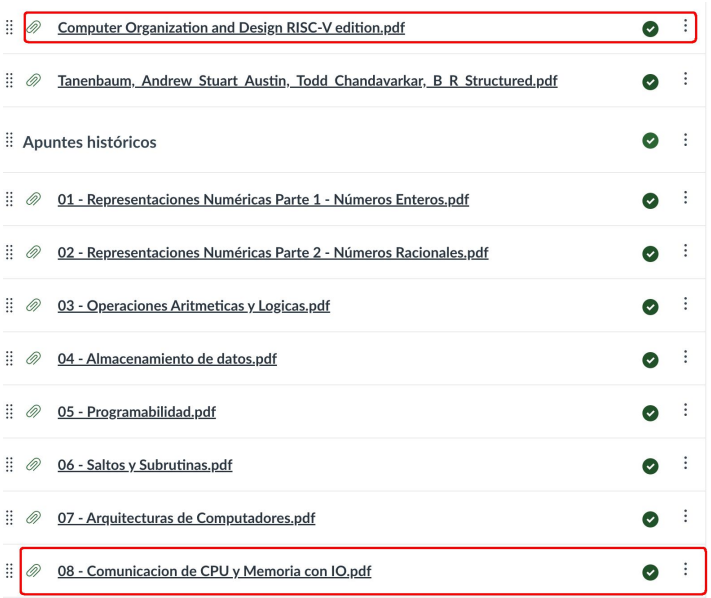
- `step_four` **(1.5 ptos.)**: Se configura un nivel alto de agua en la lavadora e inicia el lavado. Una vez que termina se abre la puerta, se expulsa la ropa (para que quede en la catapulta) y se apaga.
- `step_five` **(0.5 ptos.)**: Se lanza el contenido de la catapulta y se reposiciona.
- `step_six` **(1.5 ptos.)**: Se cierra la puerta de la secadora, se configura para toallas e inicia el secado. Una vez que termina se abre la puerta, se expulsa la ropa (para que quede en el canasto) y se apaga.

Bibliografía

- Apuntes históricos. Hans Löbel, Alejandro Echeverría

  8 - Comunicación de CPU y Memoria con IO

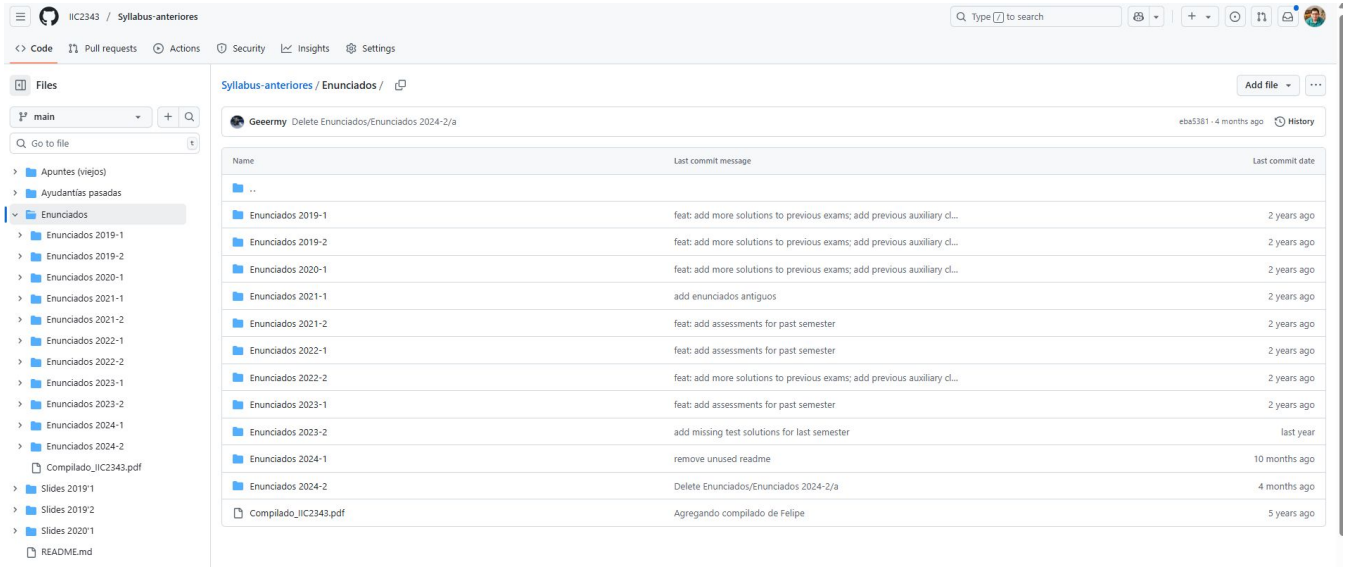
- D. Patterson, Computer Organization and Design RISC-V. Edition: The Hardware Software Interface. Morgan Kaufmann, 2020.Capítulo 6.9. Página 529.e1, 690 en PDF



¿Dudas?

Bibliografía

- https://github.com/IIC2343/Syllabus-anteriores/tree/main/Enunciados



Clase 16 - Entradas y Salidas: Comunicación

IIC2343 - Arquitectura de Computadores

Profesor:
- Felipe Valenzuela González
Correo:
frvalenzuela@alumni.uc.cl