Dispositivos de entrada/ salida

Sistemas con Microprocesadores

Mg.Ing. Esteban Volentini (evolentini@herrera.unt.edu.ar)

http://www.microprocesadores.unt.edu.ar/procesadores

Cronograma

Actividad	Inicio	Descripción	Completa
Presentación	24/08	Presentación y reglamento	✔
Tema 1	24/08	Estructura de las computadoras	✔
Tema 2	26/08	Proyecto con un microcontrolador	✔
Tema 3	31/08	Descripción funcional del ISA RV32IM	✔
Tema 4	07/09	Programación en lenguaje ensamblador	✔
Tema 5	18/09	Diseño del camino de datos de un procesador	✔
Tema 6	30/09	Diseño de la unidad de control del procesador	✔
Tema 7	05/10	Tecnologias y sistema de memoria	✔
Tema 8	09/10	Dispositivos de entrada/salida	←
Tema 9	19/10	Entradas/salidas digitales (GPIO)	
Tema 10	28/10	Generalidades de las interrupciones	
Tema 11	04/11	Interrupciones y excepciones en RISC-V	
Tema 12	13/10	Manejo de señales con temporizadores	
Tema 13	18/11	Transferencias de datos	
Proyectos	30/11	Seminarios de proyectos	

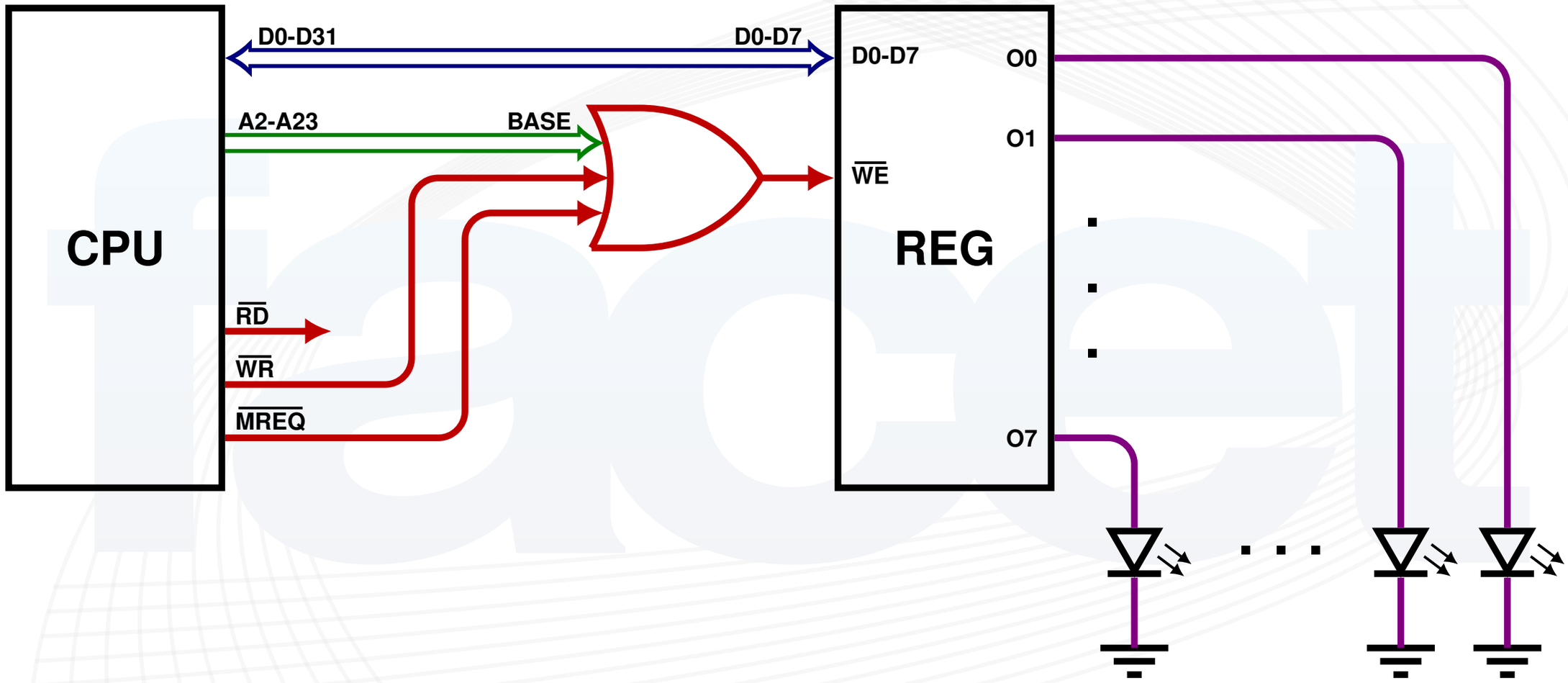
Temas a tratar

- Entradas/Salidas digitales de propósito general (GPIO)
  - Uso de un registro para salidas
  - Uso de un buffer tri-state para entradas
  - Entradas/Salidas configurables por software
- Uso de un contador como temporizador
  - Conexión del contador al bus
  - Uso de un registro para control del contador
  - Uso de buffer 3state para estado del contador
  - Generación de un pedido de interrupción
- Registros de datos, control y estado

Salidas digitales en un CPU

- Se propone agregar un conjunto de salidas digitales de propósito general
  - Por ejemplo para controlar los display de 7 segmentos en el reloj del proyecto integrador
- Se puede utilizar un registro tipo D conectado al bus
  - Se puede considerar como una memoria de un byte, es decir que ocupa solo 1 dirección
  - Se debe asignar una dirección de memoria a este registro con una lógica de selección en una dirección arbitraria **BASE**
- Las salidas del registro se conectan a los elementos que se desean controlar

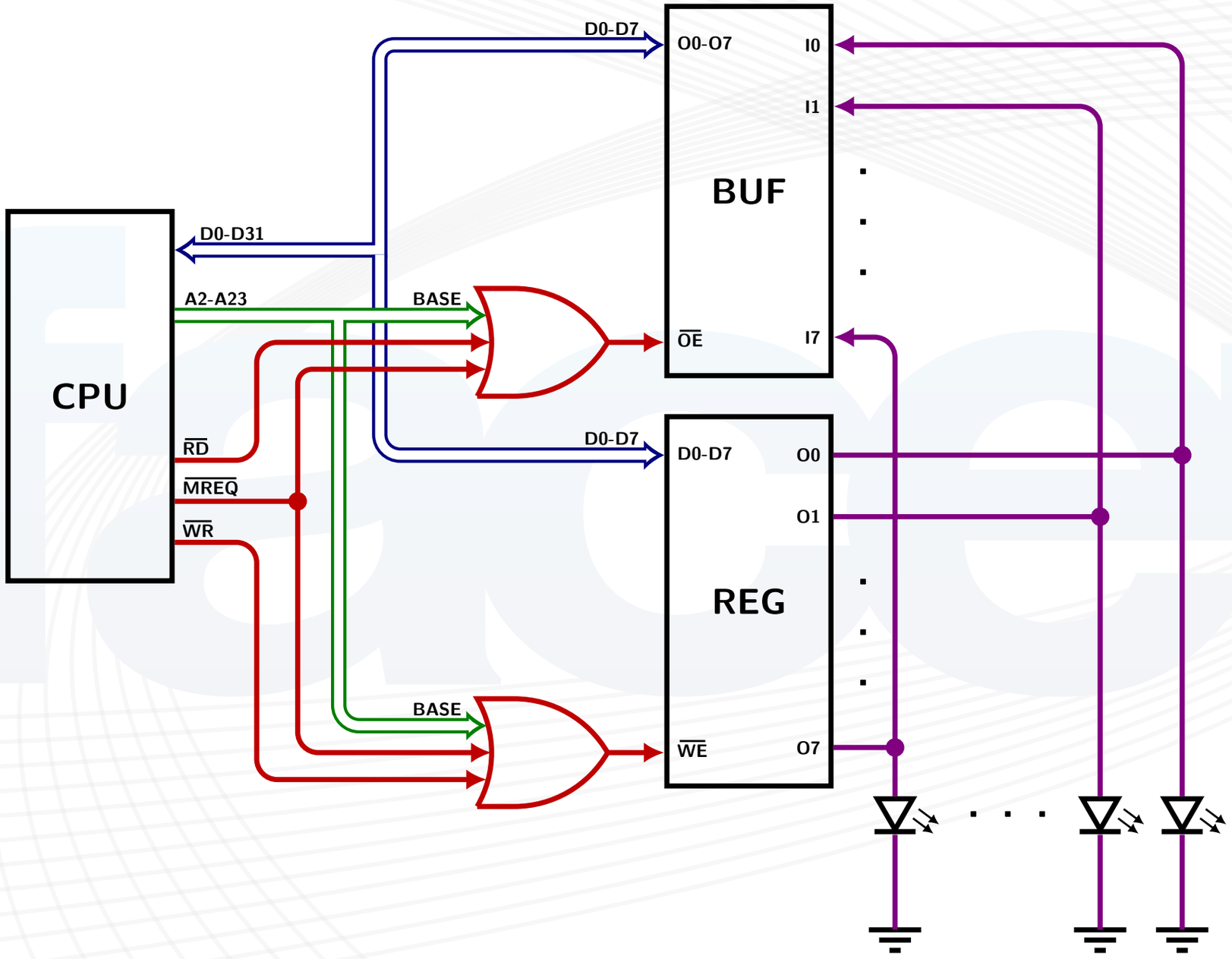
Salidas digitales en un CPU



Salidas digitales en un CPU

- Esta conexión permite controlar el estado de los leds escribiendo el valor adecuado en la dirección *BASE*
  - En este caso los leds se prenden al escribir un 1 en el bit correspondiente
  - La corriente de salida del registro debe ser suficiente para alimentar a la carga
- Sin embargo no es posible leer el estado de una salida
  - Al realizar una lectura de la dirección *BASE* no se activa ningún dispositivo y el valor resultante es indeterminado
- Esto es un problema bastante significativo cuando se desea cambiar el estado de un led sin afectar el resto
  - Sería necesario disponer de una variable para recordar el estado

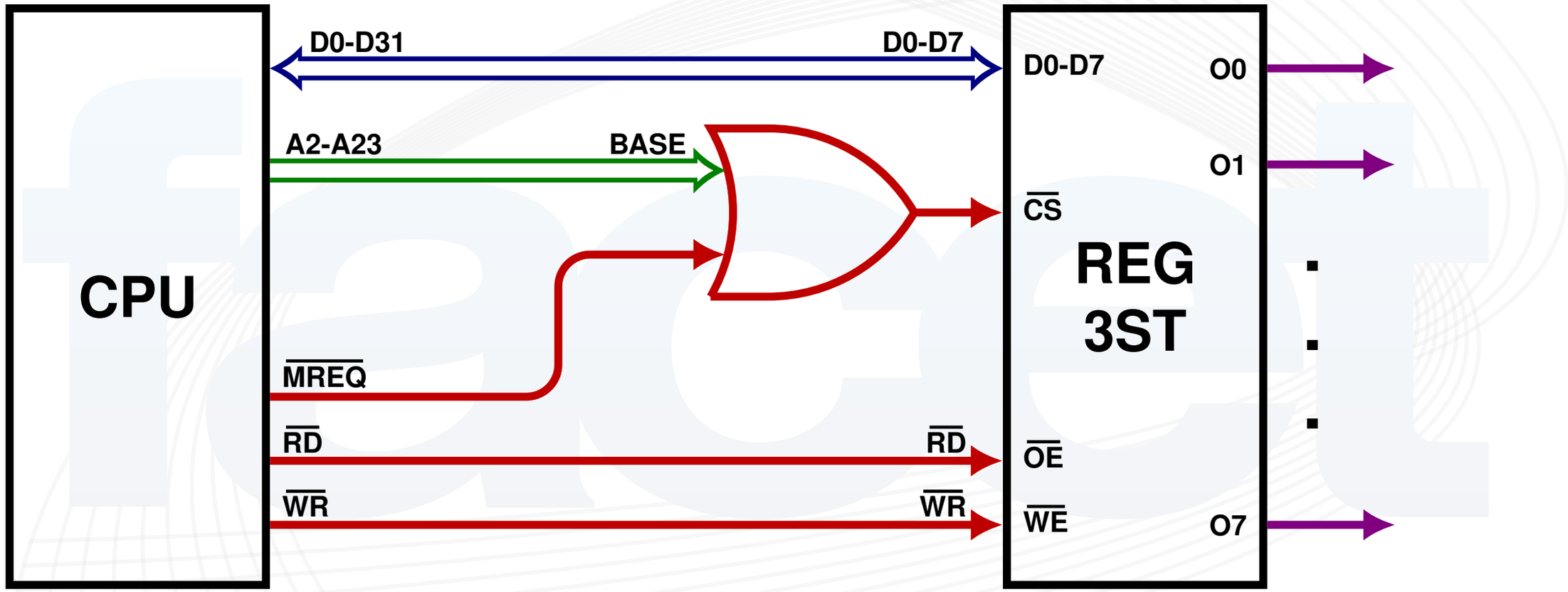
Salidas digitales en un CPU



Salidas digitales en un CPU

- Se utiliza un registro con un buffer tri-state
- Permite escribir y leer el valor del registro como una dirección de memoria normal
  - Mismos componentes que una memoria RAM estática
- Pero el contenido del los flip-flips que forman las celdas de almacenamiento están conectados a un circuito externo
  - Se puede controlar el circuito externo en función de lo que se escriben en esos flips-flops

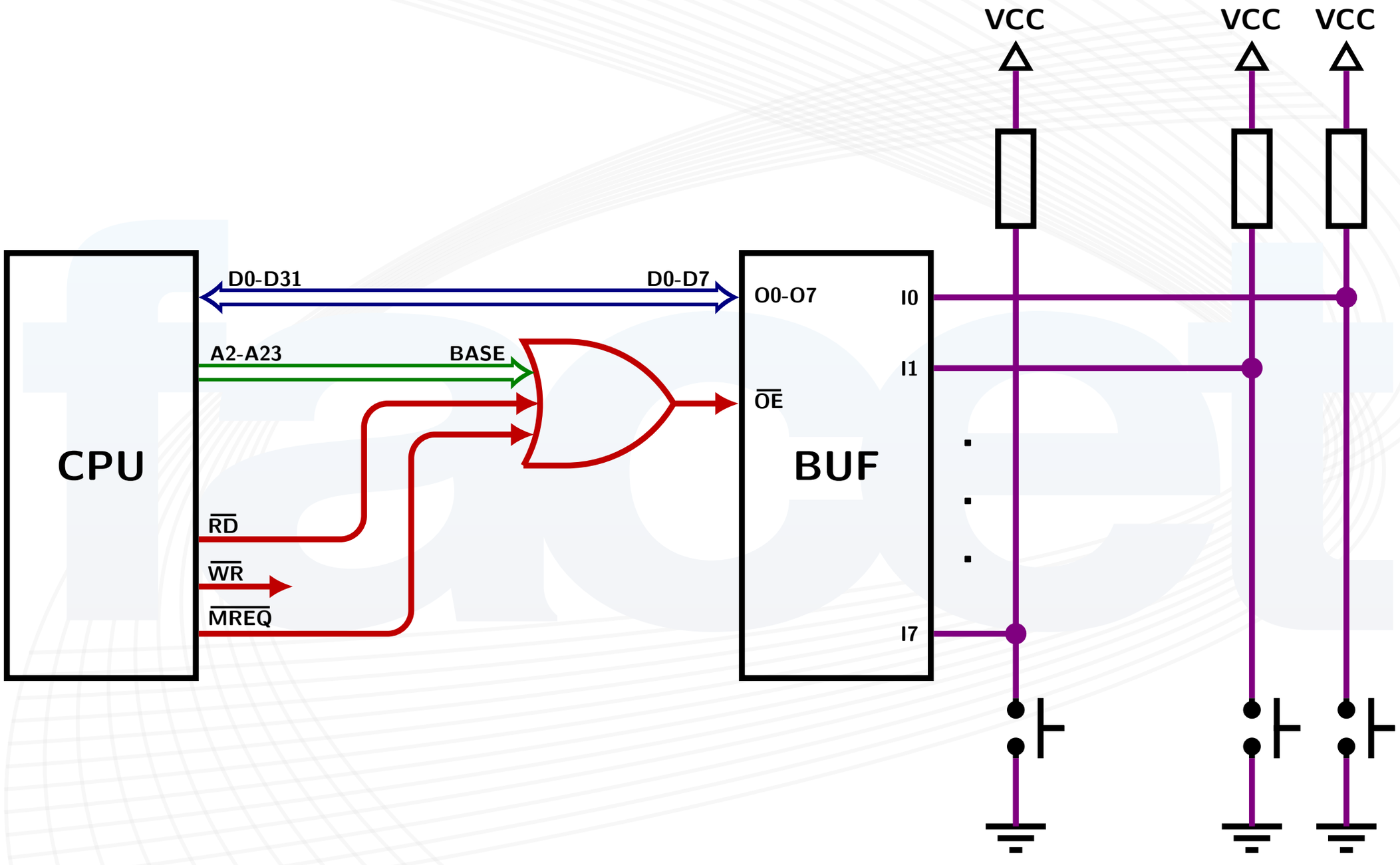
Salidas digitales en un CPU



Entradas digitales en un CPU

- Para leer el valor de entradas digitales desde un CPU se puede usar el bus de datos
- Sin embargo para conectar un dispositivo al bus es necesario que:
  - El dispositivo tenga salidas con capacidad de alta impedancia (tri-state)
  - Las salidas deben permanecer en alta impedancia hasta que se realice una lectura en una dirección específica
- La solución es utilizar un buffer tri-state que se habilita en una dirección específica **BASE**

Entradas digitales en un CPU



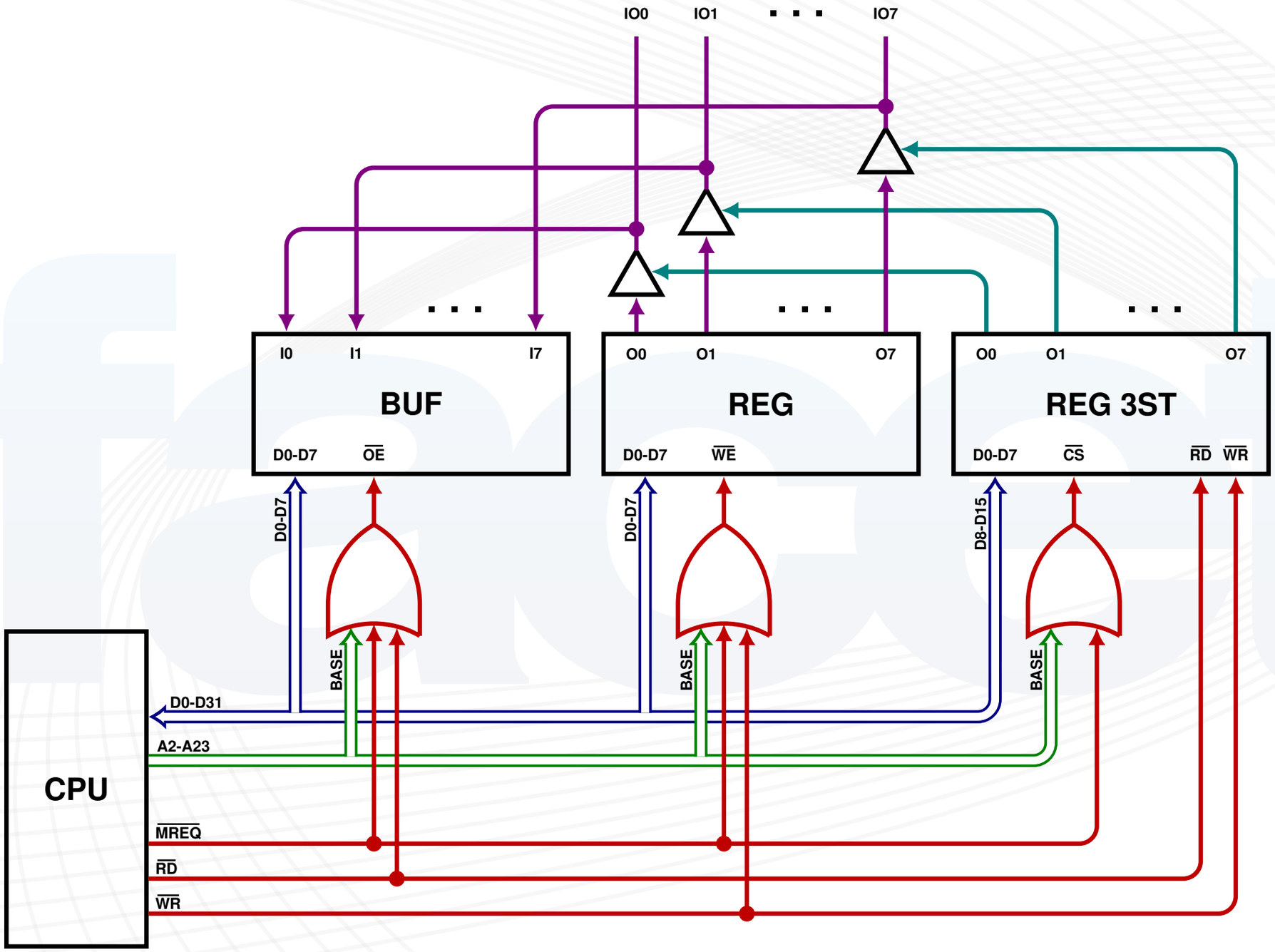
Entradas digitales en un CPU

- La entrada del buffer tri-state debe tener siempre un valor definido
  - Cuando se conectan botones se requieren resistencias de pull-up (o de pull-down)
- Esta resistencia fija el valor de entrada cuando el botón está en reposo
  - Y limita la corriente cuanto el botón está activo
- Lo mismo sucede si se utiliza un dispositivo electrónico con salida a colector abierto

Entradas/Salidas digitales (GPIO)

- Cada linea se configura como entrada o salida en forma independiente
- Es una combinación de los dos circuitos anteriores
  - Se deben agregar buffers para conectar o desconectar las salidas de los registros de los terminales
  - Se debe agregar un registro para controlar estos buffers
- Este segundo registro es el que permite configurar la dirección como salida o entrada

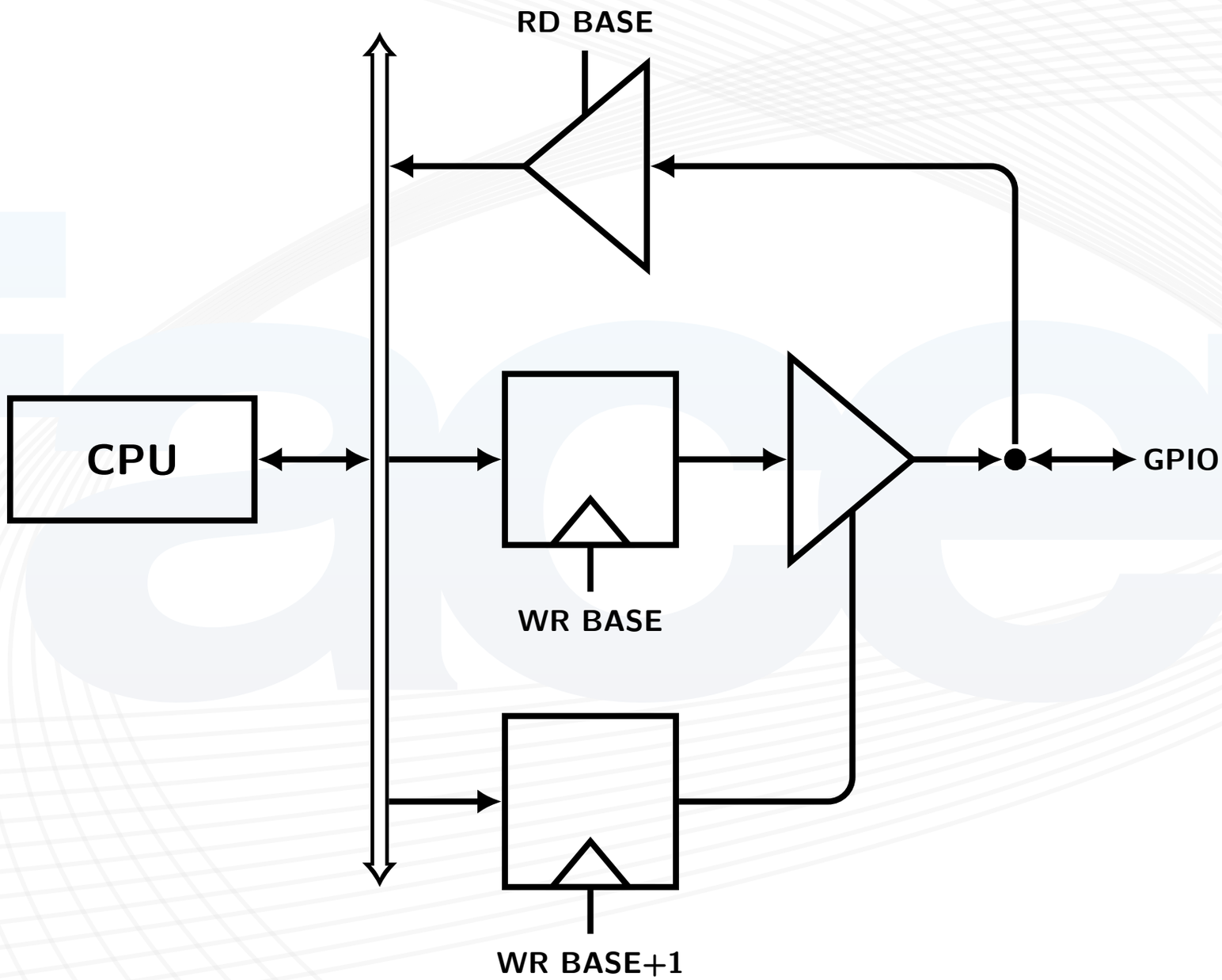
Entradas/Salidas digitales (GPIO)



Entradas/Salidas digitales (GPIO)

- Este dispositivo ocupa dos direcciones de memoria
- En la dirección **BASE** opera con las entradas o salidas
  - Se puede escribir el estado de las salidas
  - Se puede leer el estado de las entradas o de las salidas
- En la dirección **BASE+1** se configura la dirección de cada linea como entrada o salida
  - Si se escribe un 0 la línea opera como entrada
  - Si se escribe un 1 la línea opera como salida
  - Tambien se puede leer la configuración actual de las líneas
- Como el bus es de 32 bits no existen **A0** ni **A1**
  - En realidad se decodifica la misma dirección **BASE** y se utilizan las líneas de datos del segundo byte (**D8** a **D15**)

GPIO simplificado



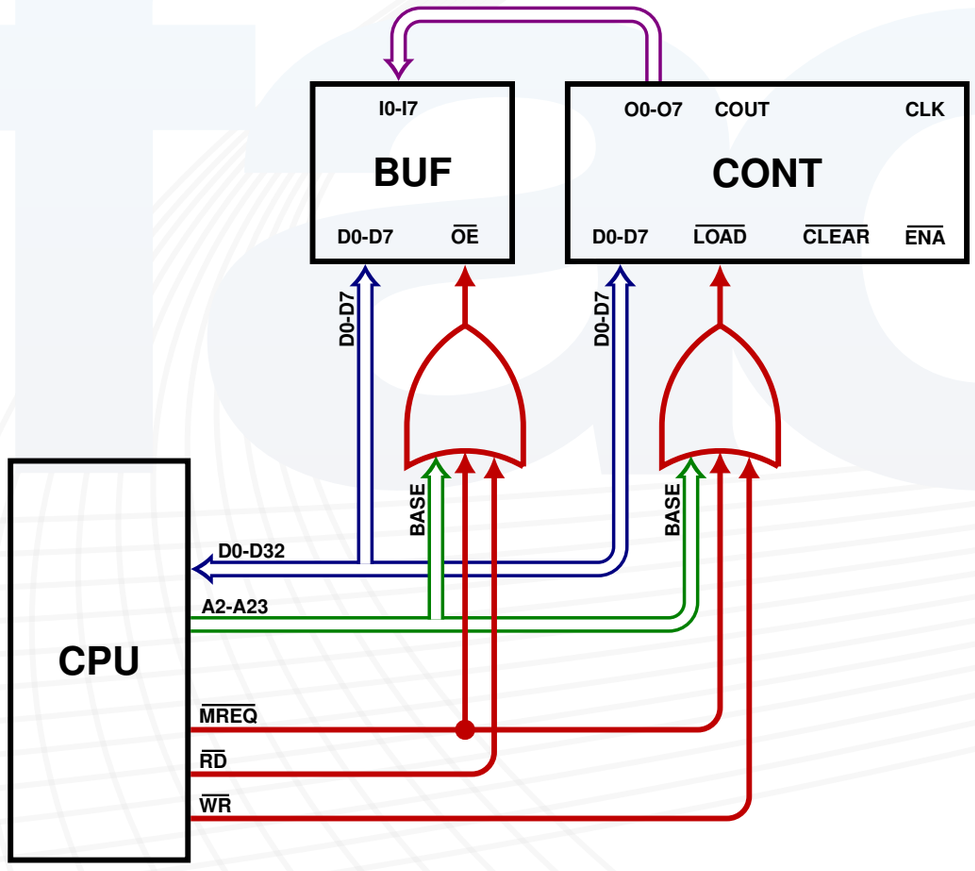
Dispositivos en un CPU

- Con entradas y salidas digitales se puede controlar cualquier dispositivo
- Por ejemplo se desea conectar un contador con las siguiente funcionalidades:
  - Seleccionar la fuente de reloj entre dos opciones
  - Iniciar o suspender la cuenta
  - Leer el valor actual de la cuenta
  - Cargar un valor inicial en el contador
  - Volver la cuenta a cero

Conexión de un contador a un CPU

- La lectura y escritura de un valor en el contador es igual que en un registro
  - Se debe generar la señal de carga a partir de la decodificación de una dirección de memoria **BASE**
  - Se debe utilizar un buffer tri-state para que las salidas del contador se lean desde el bus
    - Este buffer se habilita con la decodificación de la misma dirección **BASE** utilizada para la carga del contador
- Para el software es como una dirección de memoria normal que puede leerse y escribirse

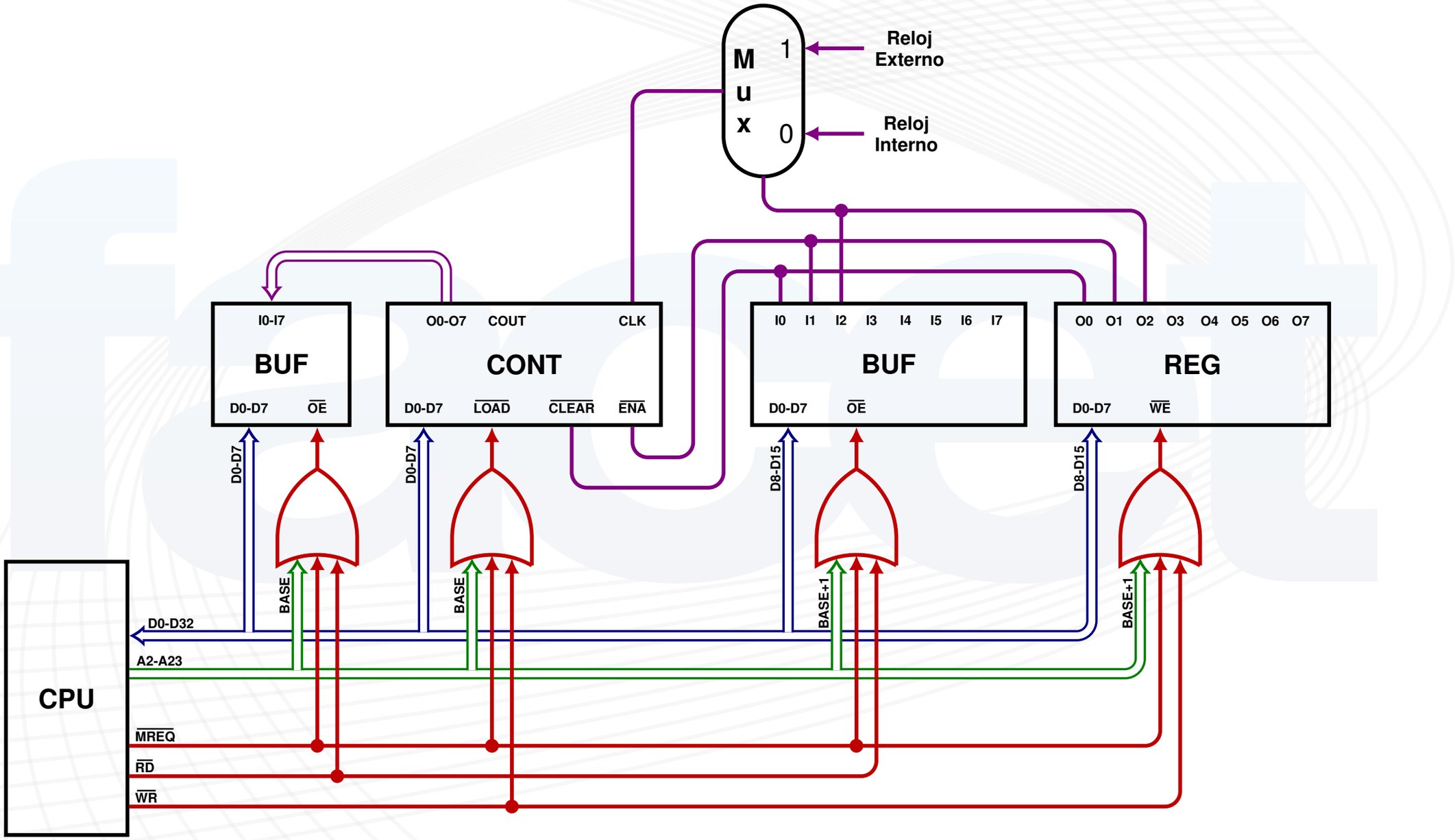
Conexión de un contador a un CPU



Conexión de un contador a un CPU

- Para manipular las señales de control del contador se necesitará un registro adicional
  - Este registro se mapea, conceptualmente, en una segunda dirección de memoria **BASE + 1**
  - Se utilizarán señales de este registro para activar las señales de borrado y habilitación del contador
  - Se agregará un multiplexor para seleccionar la fuente de reloj del contador
    - Se controla tambien con una señal del nuevo registro
  - Al registro se le agrega un buffer para poder leerlo

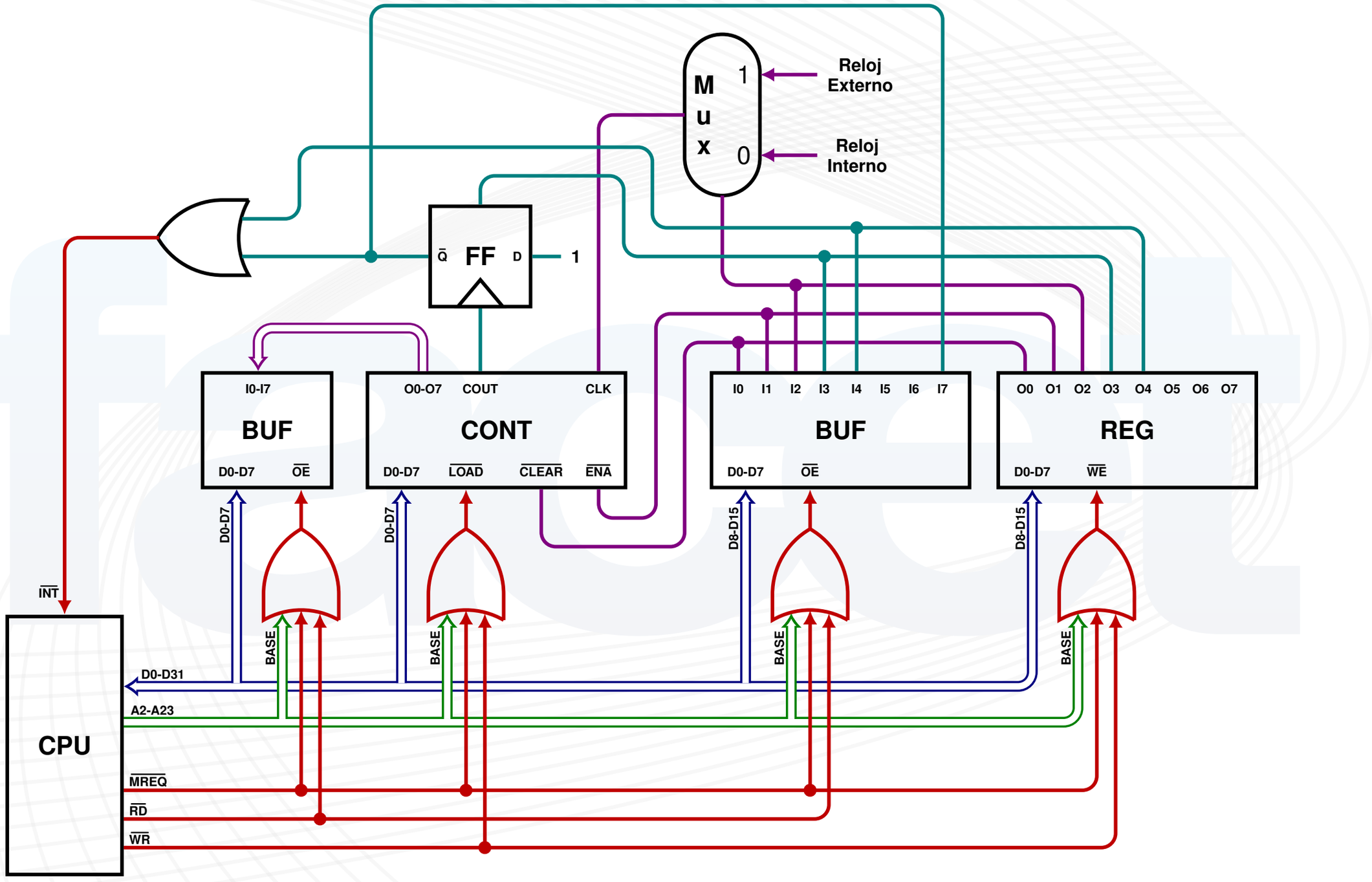
Conexión de un contador a un CPU



Conexión de un contador a un CPU

- Agregar al contador la capacidad para generar un pedido de interupción en el CPU
  - El procesador tiene una entrada específica para este propósito
  - El pedido de interrupción se debe generar cuando el contador alcance la cuenta máxima
    - Se utiliza la señal carry out del contador
    - Se debe memorizar hasta que el pedido sea atendido
  - Se debe poder consultar si hay una interrupción pendiente, borrarlo y enmascararlo por software

Conexión de un contador a un CPU



Conexión de un contador a un CPU

- El contador ocupa dos direcciones de memoria
  - En la dirección **BASE** se encuentra el registro de datos
  - En la dirección **BASE + 1** se encuentra el registro de control y estado
- Se puede operar sobre el contador con operaciones de lectura y escritura en estas direcciones
  - Sin embargo conocer estas direcciones no es suficiente para escribir un programa adecuado
  - Es necesario conocer la función de cada uno de los bits del registro de control y estado

Registro de control y estado

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
STATUS	N/I	N/I	MASK	ACK	CLOCK	ENABLE	CLEAR

- CLEAR: bit de borrado
  - CLEAR = 0 → El contador mantiene su valor
  - CLEAR = 1 → El contador se borra asincronicamente
- ENABLE: bit de habilitación de cuenta
  - ENABLE = 0 → El contador está detenido
  - ENABLE = 1 → El contador está habilitado y contando
- SEL_CLK: bit de selección de clock
  - SEL_CLK = 0 → El contador opera con un clock interno
  - SEL_CLK = 1 → El contador opera con un clock externo

Registro de control y estado

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
STATUS	N/I	N/I	MASK	ACK	CLOCK	ENABLE	CLEAR

- ACK: bit de reconocimiento del evento de overflow
  - ACK=1 → Borra el pedido bit de STATUS y el pedido de interrupción
  - ACK=1 → Se memoria el evento de overflor en el bit de STATUS
- MASK: bit de máscara de interrupción por overflow
  - MASK = 1 → Las interrupciones están inhabilitadas
  - MASK = 0 → Las interrupciones están habilitadas
- STATUS: bit de status de overflow
  - STATUS = 0 → No ocurrió overflow en el contador
  - STATUS = 1 → Ocurrió un overflow en el contador

Rutinas de control del contador

```
start:
    la t0,cont          # Se apunta al registro base del contador
    li t1,0x00          # Borrar e inhabilitar el contador
    sb t1,1(t0)         # Se actualiza el registro de control
    li t1,0x06          # Habilitar el contador con reloj externo
    sb t1,1(t0)         # Se actualiza el registro de control
    ret

stop:
    la t0,cont          # Se apunta al registro base del contador
    lbu t1,1(t0)        # Se carga el valor actual del registro
    andi t1,t1,0xFD     # Inhabilitar el contador
    sb t1,1(t0)         # Se actualiza el registro de control
    lbu a0,0(t0)        # Se leer el valor actual del contador
    ret
```

Conclusiones

- Se conectaron dispositivos de entrada/salida al procesador como si fueran memoria
  - Para el programa no hay diferencia
- Un dispositivo tiene más de un tipo de registro
  - Registros de datos, de control y de estado
- Se necesita conocer la dirección de memoria y la función de cada bit de estos registros
  - El mapa de memoria tiene mas detalle cuando se trata de los dispositivos de entrada/salida