



Práctica 6

Modificaciones examen

MIPS multicycle



Control de excepciones [0,5 puntos]

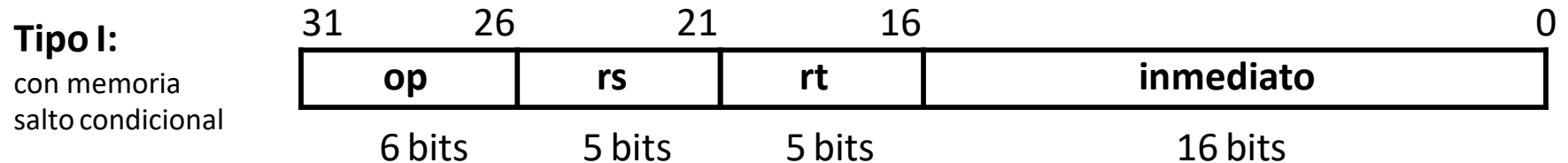
- **Si en el estado S3 el código de operación es desconocido, transitar a un nuevo estado de excepción**
 - Permaneceremos en este nuevo estado de excepción de manera permanente y se encenderá uno de los leds de la barra de leds.
- **Probar en FPGA que se hace control de excepciones con código de operación desconocido**
 - Cambiar el código de operación de alguna de las instrucciones por un con código de operación inexistente (por ejemplo, "111111") o utilizar el programa proporcionado (examenExcepcion.coe)

Instrucción jal (salto incondicional relativo a PC) [0,5 puntos]



■ Añadir instrucción jal:

– jal dir $PC \leftarrow PC + 4 + 4 * \text{SignExt}(\text{inmed})$



- Código de operación: “000011”

■ Probar en FPGA que el resultado mostrado por los displays es correcto para los valores introducidos (probar con valores 2 y 5)

- Utilizar el programa proporcionado (examenJAL.coe)
 - Sigue siendo el mismo algoritmo de multiplicación mediante sumas iterativas de casa que incluye esta nueva instrucción

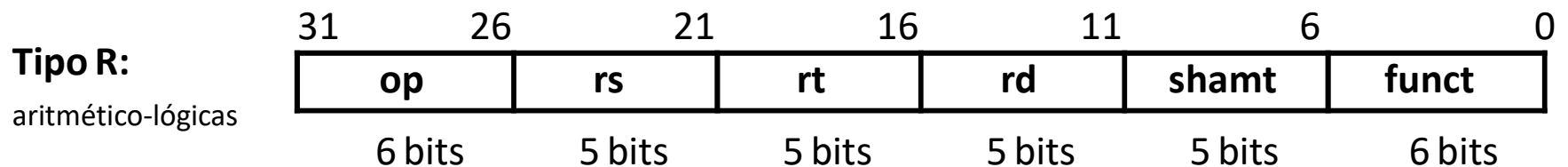
Instrucciones de desplazamiento

[0,5 puntos]



■ Añadir instrucción desplazamiento a la derecha:

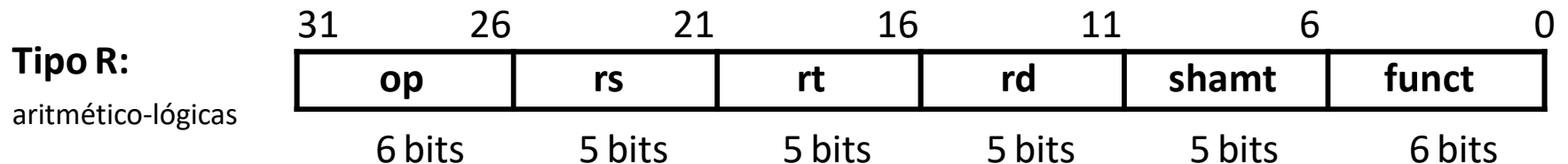
– sra1 rd, rs $rd \leftarrow rs \gg 1, PC \leftarrow PC + 4$



- Desplaza un bit a la derecha y replica el MSB (despla. aritmético)
- funct: “000000”

■ Añadir instrucción desplazamiento a la izquierda:

– sll1 rd, rs $rd \leftarrow rs \ll 1, PC \leftarrow PC + 4$



- Desplaza un bit a la izquierda y pone un 0 en el LSB (despla. lógico)
- funct: “000010”

Instrucciones de desplazamiento

[0,5 puntos]



- **Probar en FPGA que el resultado mostrado por los displays es correcto para los valores introducidos (probar con valores 2 y 5)**
 - Utilizar el programa proporcionado (examenSHIFT.coe)
 - En este caso es el algoritmo de multiplicación mediante sumas iterativas que utilizaba desplazamientos

Instrucción mostrar en la barra de leds los 8 LSB de un registro [1 punto]

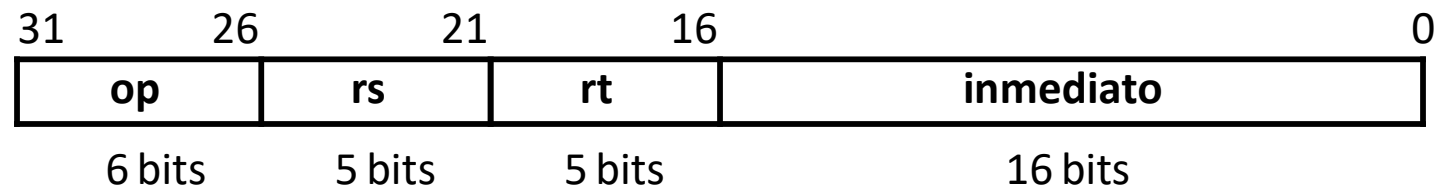


- **Añadir instrucción para mostrar el contenido de un registro en ese instante en la barra de leds:**

– `mrl rs leds <- rs[7:0], PC <- PC + 4`

Tipo I:

con memoria
salto condicional



- Código de operación : “111100”

- **Probar en FPGA que coincide el resultado mostrado por los displays y en la barra de leds para los valores introducidos (probar con valores 2 y 5)**

- Utilizar el programa proporcionado (`examenMRL.coe`)
 - Sigue siendo el mismo algoritmo de multiplicación mediante sumas iterativas de casa que incluye esta nueva instrucción