



Práctica 6

Modificaciones parte de casa

MIPS multicycle

Mostrar R3 por displays 7 segmentos



- **Mostrar por los displays 7 segmentos de la placa extendida el contenido del registro R3**
 - Este registro es el que utilizaremos en nuestros programas para almacenar el resultado
 - Se mostrarán los 4 bits menos significativos en el display de menor peso y los 4 bits siguientes en el otro display

Aclaración: No se trata de crear una nueva instrucción, simplemente hay que mostrar el contenido del registro R3 por los displays de 7 segmentos.

- **Probar en FPGA que el resultado mostrado por los displays es correcto**

Modo depuración



- **Añadir la funcionalidad de depuración instrucción a instrucción:**
 - Mostrar en el display 7 segmentos de la placa superior el resultado de $((PC - 4) \text{ módulo } 4)$ ó $(PC[31:2] - 1)$
 - Mediante uno de los SW de la placa extendida se seleccionará el modo de funcionamiento:
 - Normal: el programa se ejecuta como hasta ahora instrucción a instrucción sin esperas
 - Depuración: el programa se detiene al comienzo de cada instrucción (estado S1) y permanece así hasta que se presione un pulsador
- **Probar en FPGA que el resultado mostrado por los displays es correcto**

Instrucción lectura switches

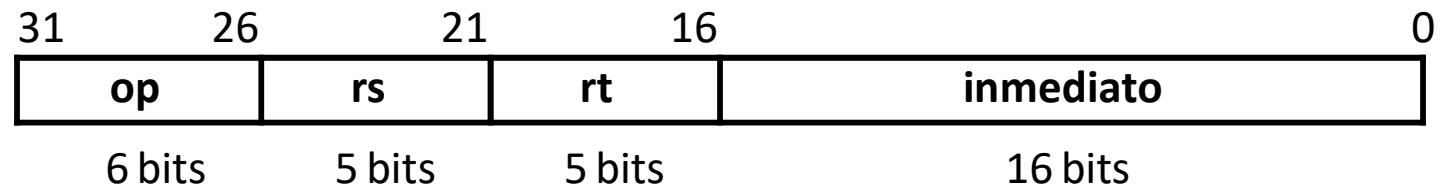


■ Añadir instrucción lectura switches (SW):

- lsw rt, #inmed si (inmed = 0) $rt \leftarrow \text{SignExt}(\text{SW placa extendida}[3:0])$, $PC \leftarrow PC + 4$
 en otro caso $rt \leftarrow \text{SignExt}(\text{SW placa superior}[3:0])$, $PC \leftarrow PC + 4$

Tipo I:

con memoria
salto condicional



- Código de operación: “010001”

- **Modificar el programa dado para que un operando se lea del SW superior y el otro del SW de la placa extendida**
- **Probar en FPGA que el resultado mostrado por los displays es correcto para los valores introducidos**

Instrucción move

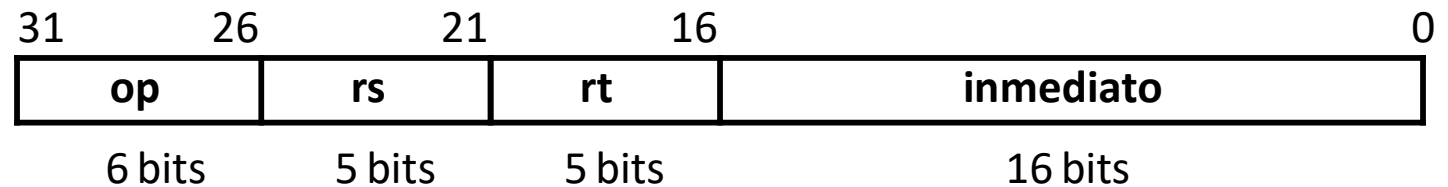


■ Añadir instrucción move con inmediato:

– mv rt, #inmed $rt \leftarrow \text{SignExt}(\text{inmed}), PC \leftarrow PC + 4$

Tipo I:

con memoria
salto condicional



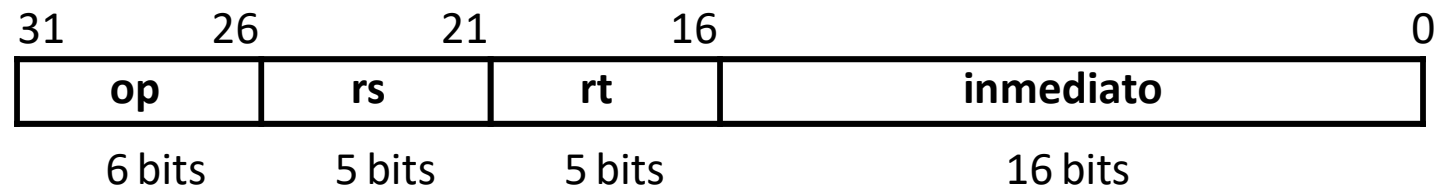
- Código de operación: “010000”

■ Añadir instrucción move con registro:

– mv rt, rs $rt \leftarrow rs, PC \leftarrow PC + 4$

Tipo I:

con memoria
salto condicional



- Código de operación: “010010”

Instrucción move



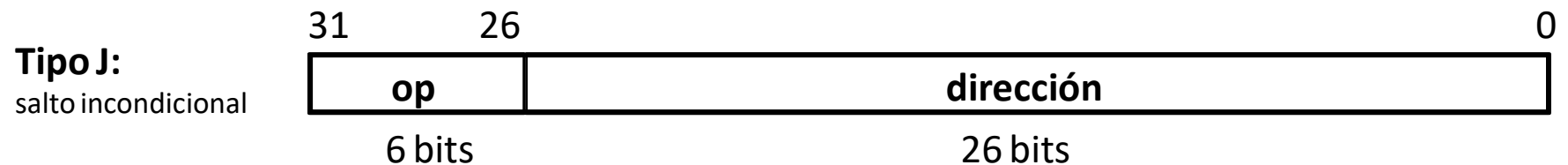
- **Modificar el programa dado utilizando la instrucción move para poner a 0 y 1 los registros**
 - 1) El registro R3 se pondrá a 0 con inmediato
 - 2) El registro R4 se pondrá a 0 copiando el registro R3
 - 3) El registro R2 se pone a 1 con inmediato
- **Probar en FPGA que el resultado mostrado por los displays es correcto para los valores introducidos**

Instrucción jump (salto incondicional)



■ Añadir instrucción jump:

— j dir PC <- "000000"&dir



- Código de operación: "000010"

- **Modificar el programa dado utilizando la instrucción de salto incondicional**
- **Probar en FPGA que el resultado mostrado por los displays es correcto para los valores introducidos**