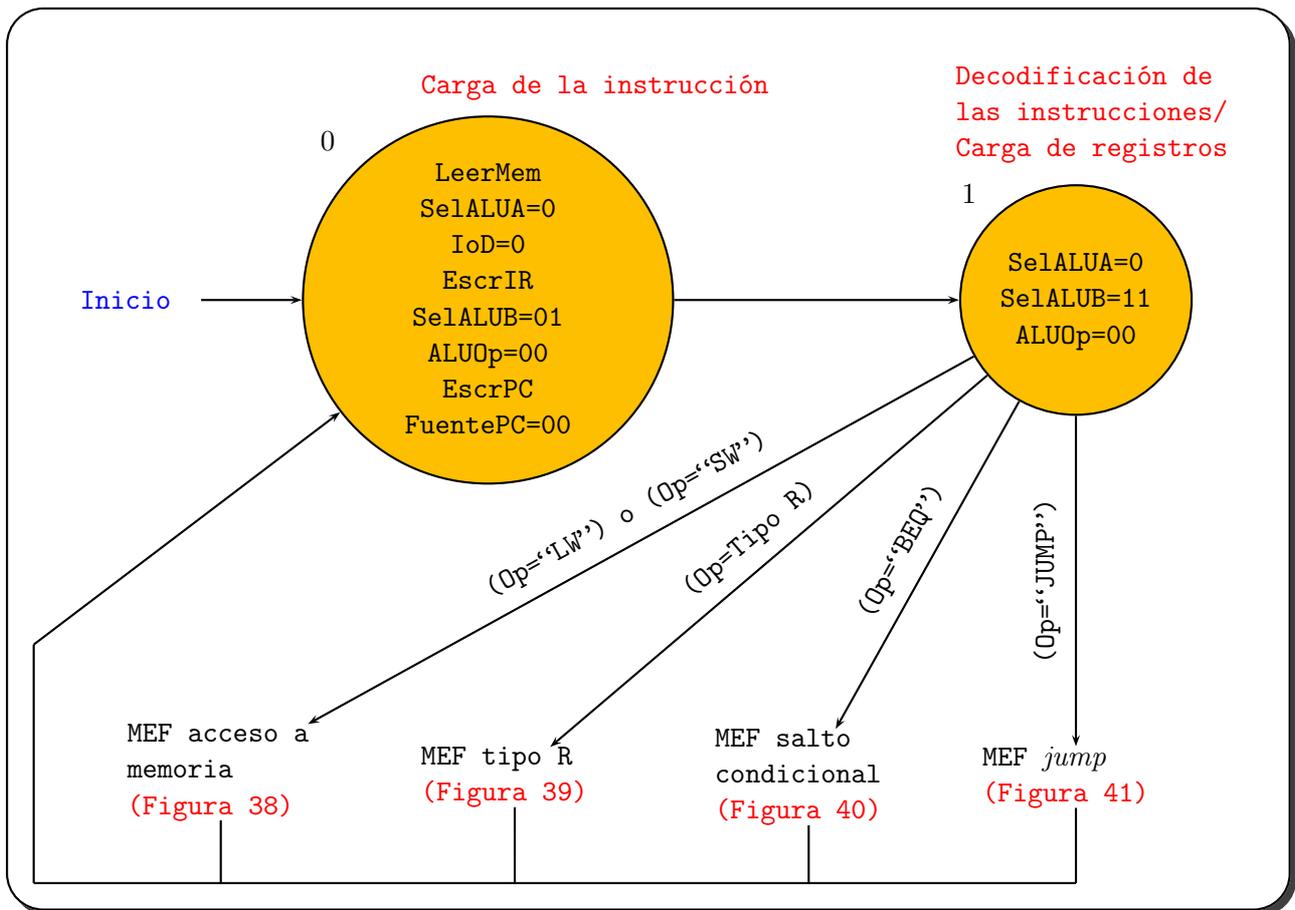
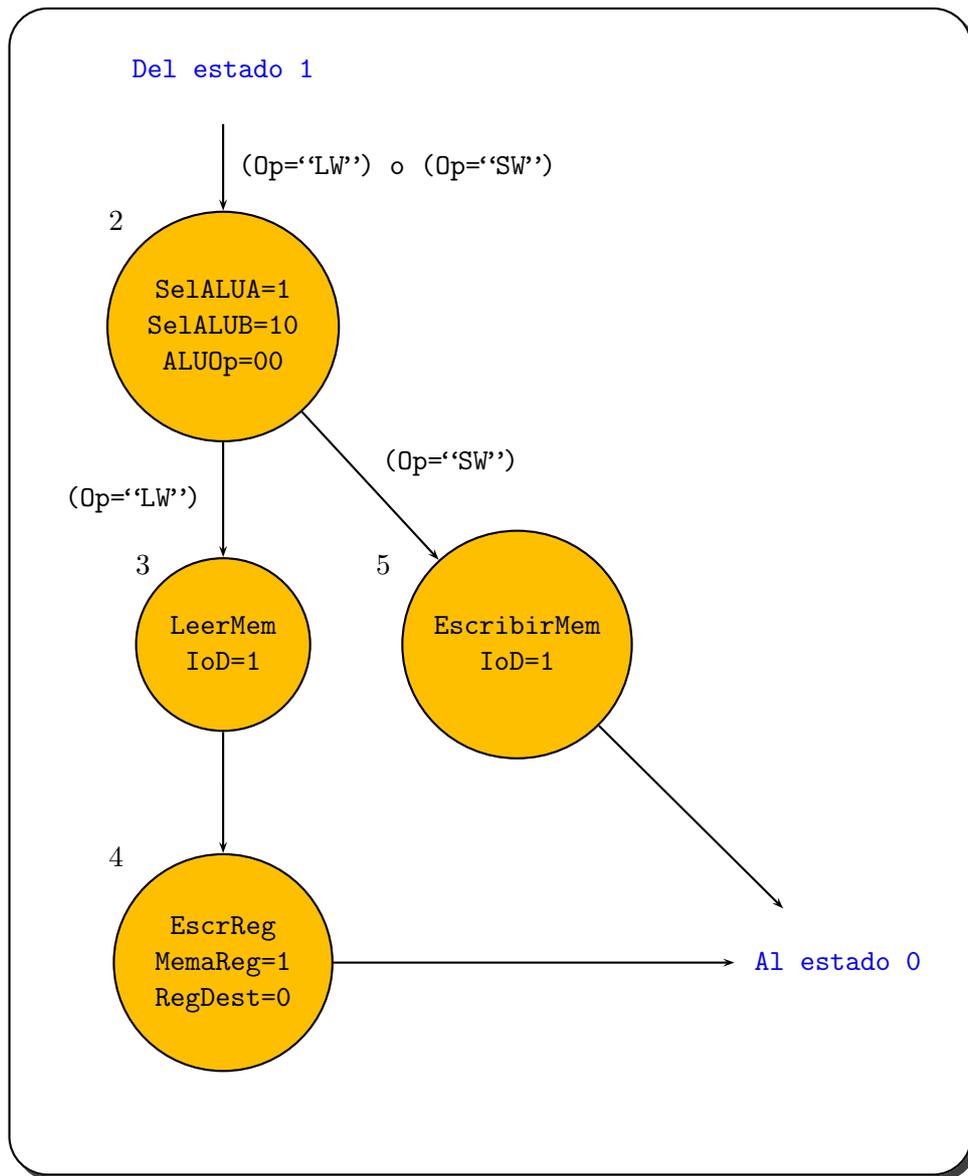


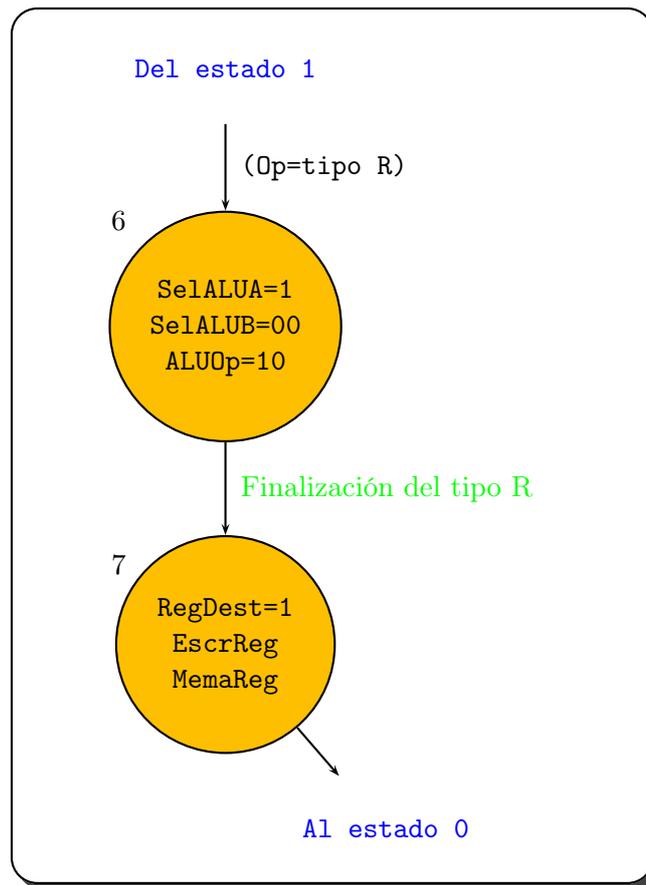
Figura 5.36.- Una visión de alto nivel del control mediante una máquina de estados finitos (MEF). Las primeras etapas son independientes del tipo de instrucción. Después, para completar cada tipo, se utiliza una serie de secuencias que dependen del código de operación. Tras la finalización de todas las acciones requeridas por cada instrucción, el control debe volver a cargar una nueva. Cada caja de la figura puede representar más de un estado. El arco etiquetado como inicio señala el estado donde debe comenzarse cuando se vaya a buscar la primera instrucción a memoria. (5.36)



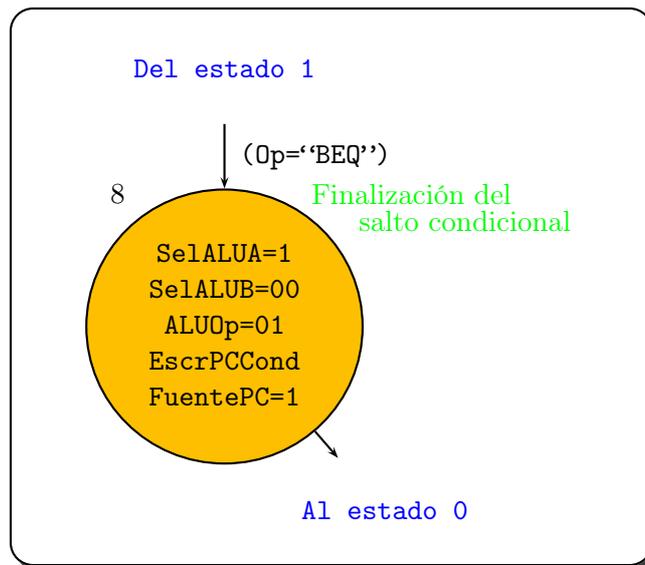
La búsqueda de la instrucción y la fase de decodificación son idénticas para todas las instrucciones. Estos estados corresponden a la caja superior de la máquina de estados finitos abstracta de la figura 5.36. En el primer estado, se activan dos señales para ir a leer la instrucción a memoria y a escribirla en el Registro de Instrucción (LeerMem y EscrIR), y se pone IoD a 0 para poder elegir el PC como fuente de la dirección. A las señales SelALUA, SelALUB, ALUOp, EscrPC y FuentePC se les asocian valores para calcular PC+4 y guardarlo en el PC (también se almacena en SalidaALU, pero no se utiliza nunca). En el siguiente estado, se calcula la dirección destino del salto poniendo SelALUB a 11 (los 16 bits de menor peso del IR desplazados 2 bits a la izquierda y con el signo extendido se envían a la ALU), poniendo SelALUA a 0 y ALUOp a 00; el resultado se guarda en el registro SalidaALU, el cual se escribe en cada ciclo. Los siguientes cuatro estados dependen del tipo de instrucción, el cual se conoce en este estado. La entrada de la Unidad de Control, llamada Op, se utiliza para determinar qué arco debe seguirse. (5.37)



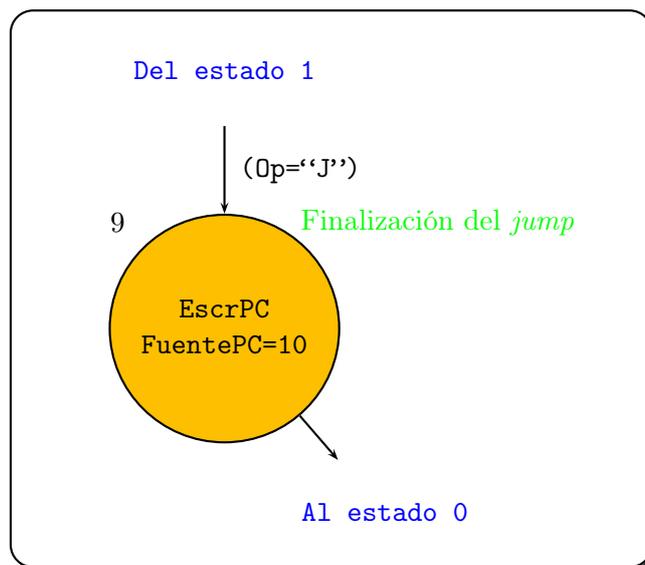
La máquina de estados finitos para controlar las instrucciones de acceso a memoria tiene cuatro estados. Estos estados corresponden a la caja etiquetada como “instrucciones de acceso a memoria” de la figura 5.36. Tras calcular la dirección de memoria, se necesitan dos secuencias diferentes para la ejecución de *loads* y *stores*. La activación de las señales SelALUA, SelALUB y ALUOp está hecha de manera que se calcula la dirección a acceder en el estado 2. Las instrucciones *load* requieren de un estado extra para escribir el resultado desde el MDR (donde se escribe el resultado en el estado 3) dentro del banco de registros. (5.38)



Las instrucciones de tipo R pueden realizarse mediante una máquina de estados finitos sencilla de sólo dos estados. Estos estados se corresponden a la caja etiquetada como “instrucciones de tipo R” de la figura 5.36. El primer estado se encarga de que se realice la operación de la ALU, mientras que el segundo se encarga de que el resultado de la operación (que está en SalidaALU) se escriba en el banco de registros. Las tres señales afirmadas durante el estado 7 son las encargadas de que el contenido del registro SalidaALU se escriba en el registro especificado por el campo rd del Registro de Instrucción. (5.39)



La instrucción de salto condicional requiere un único estado. Las primeras tres salidas que se encuentran afirmadas se encargan de que la ALU compare los registros (Se1ALUA, Se1ALUB y ALUOp), mientras que las señales de FuentePC y EscrPCCond realizan la escritura condicional del PC si la condición del salto es cierta. Obsérvese que no se usa el nuevo valor escrito en SalidaALU, en cambio se utiliza la señal de salida **Cero** de la ALU. La dirección destino del salto se lee de SalidaALU, donde se ha salvado al finalizar el estado 1. (5.40)



La instrucción *jump* requiere un único estado que activa dos señales de control para escribir en el PC los 26 bits más bajos del Registro de Instrucción desplazados 2 bits a la izquierda y concatenados con los 4 bits del PC de esta instrucción. (5.41)

