

CC4301 Arquitectura de Computadores

Solución Auxiliar 8

Prof. Aux.: Gaspar Pizarro V.

9 de noviembre de 2012

1. P3 Control 2 2008-1

1. La única forma de cargar el latch AR es haciendo una operación en la ALU y poniendo su resultado en el bus (El cable que “rodea” la CPU). Entonces no se pueden hacer las operaciones referidas en el mismo ciclo de reloj, porque en ambos casos hay que pasar por la ALU.
2. Sí, con un truco sucio: La arquitectura de M32 dispone de 32 registros, y el registro R0 siempre tiene valor cero. Entonces, si en IR hay una instrucción $XXX\ R0, R1, R2$, por Rs1 sale un cero, y se ponen las siguientes señales de control: $OP-Y-SEL \leftarrow @4$ $SEL-REG \leftarrow 1$ $WR-AR \leftarrow 1$ $OP-ALU \leftarrow @ADD$ $WR-PC \leftarrow 1$. Si no se dispusiera de R0 no se podría hacer lo pedido.
3. Sí. $OP-Y-SEL \leftarrow @0$ $WR-Rd \leftarrow 1$ $OP-ALU \leftarrow @OR$ $OP-Y-SEL \leftarrow 0$
4. $RD-DEST \leftarrow 1$ $WR-PC \leftarrow 1$ $OP-ALU \leftarrow @ADD$ $SEL-REG \leftarrow 1$
5. No, porque el MUX superior no puede dejar pasar a PC y Rs2 al mismo tiempo.
6. $OP-Y-SEL \leftarrow @INST$ $SEL-REG \leftarrow 1$ $OP-ALU \leftarrow @ADD$ $WR-IR \leftarrow 1$
7. No, porque hay un acceso a memoria, lo que implica que en el mejor de los casos la memoria responde en el ciclo de reloj siguiente al de la petición. Sin embargo, si la memoria fuera tan rápida que pudiera responder en el mismo ciclo de reloj con un valor, sí se podría, hacer, encendiendo las señales SEL-D y WR-PC.
8. No. Un argumento fácil es que como hay acceso a memoria, entonces ya no se puede hacer la operación en un solo ciclo de reloj. Sin embargo, incluso si la memoria fuera rápida (como en el punto anterior) no se podría hacer lo pedido, porque habría conflicto en el MUX inferior (pasa $R1+R3$ para entrar en Rd, o pasa Mem[AR] para entrar en IR).

2. P1 Control 3 2011-1

1. No es posible implementar SWAP en M32 como una sola instrucción¹ porque falta un registro para almacenar el valor temporal de intercambio.
2. Se propone un registro auxiliar para el valor temporal. Esto se muestra en la figura 1. Este registro es un latch, y almacenará el valor de $Mem[r_1 + \{imm|r_2\}]$, en el primer ciclo de ejecución de SWAP, y al ciclo siguiente, simultáneamente, poner el valor de Laux en r_1 y poner el valor de r_1 en $Mem[r_1 + \{imm|r_2\}]$. Además, para hacer el swap entre memoria y Laux, se propone un multiplexor en R-BANK, que elija entre la salida de Laux y el canal original de M32. El latch será controlado por la señal de control WR-L y el MUX será controlado por la señal SEL-L.

¹Una sola instrucción se refiere a hacerlo en solo un ciclo Fetch-Decode-Execute, o sea sin tener que actualizar IR ni PC. Claramente sí se pueden intercambiar dos valores en memoria o algo así (sinó poco se podría hacer en M32), pero lo que se pide es hacerlo en una sola instrucción sin tocar mas registros ni memoria que los especificados en la instrucción.

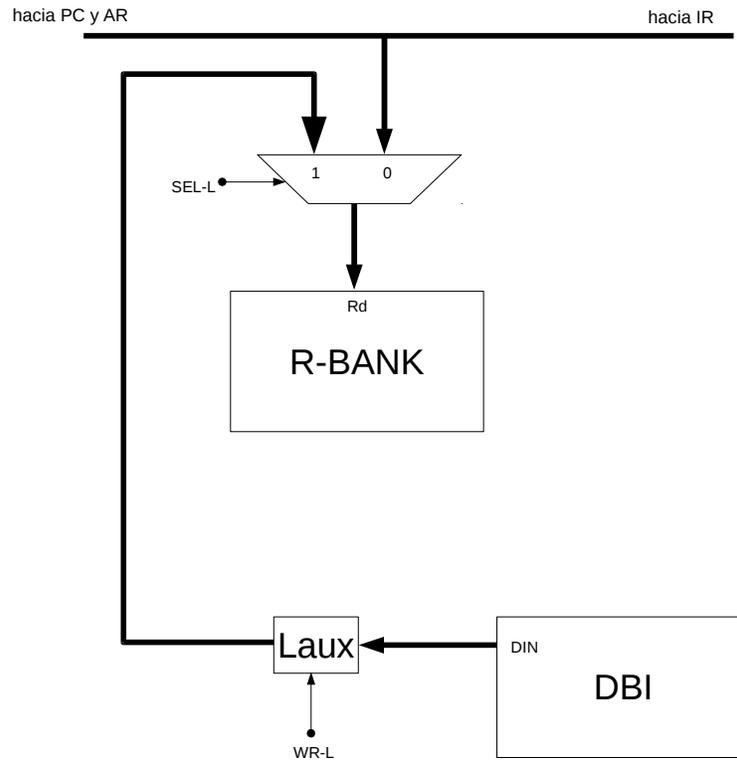


Figura 1: Modificación para implementar SWAP

3. Esto se vio en clase:

Ex1 OP-DBI \leftarrow @LDW
 OP-Y-SEL \leftarrow @INST
 SEL-D \leftarrow 0
 WR-AR \leftarrow 1
 EN-A \leftarrow 1
 WR-L \leftarrow 1

Ex2 SEL-L \leftarrow 1
 OP-DBI \leftarrow @STW
 RD-DEST \leftarrow 1
 SEL-REG \leftarrow 1
 OP-ALU \leftarrow @OR
 OP-Y-SEL \leftarrow @0
 WR-Rd \leftarrow 1

Aquí se ha hecho el supuesto de que la memoria es muy rápida, de forma que en el mismo ciclo (en Ex1) se puede cargar AR y preguntar por memoria, y usar el valor devuelto para cargar Laux. Queda propuesto hacerlo para el caso “lento”, que es que al ciclo siguiente de cargar AR, aparece por DBI el valor requerido, además que en ese ciclo, por seguridad, AR se debe mantener constante.
Hint: Hágalo en más de dos ciclos de reloj.