

IS09 - Introducción a los Computadores

Primer parcial - 07 de febrero de 2002.

Ingeniería Técnica en Informática de Sistemas.
SOLUCIONES

1. Preguntas teóricas

■ Pregunta 1

La diferencia fundamental es que la memoria principal se utiliza para almacenar los datos e instrucciones con que trabaja la Unidad Central de Proceso para ejecutar el programa en curso, mientras que la entrada/salida se utiliza para enviar y recibir datos desde los dispositivos periféricos, es decir, en un sentido no estricto, desde el exterior del computador.

■ Pregunta 2

Una instrucción de división entera como la propuesta debería devolver dos resultados. Esto presenta los siguientes problemas:

- Las instrucciones deberían poder especificar dos operandos destino, uno para el cociente y otro para el resto. Las del procesador básico visto sólo devuelven un resultado.
- La unidad aritmético-lógica debería disponer de dos registros de salida, y no de uno como es el caso del computador básico.
- Para no repetir registros de salida, se podría incluir una segunda fase de escritura de resultados, pero esto también entra en conflicto con la ejecución normal de las instrucciones en el computador básico.

■ Pregunta 3

El número 128 en binario natural se expresa 10000000, con 8 bits. Veamos el resto de formatos:

- Signo magnitud: habría que añadir un bit para el signo, teniendo $128 = 01000000$ y $-128 = 11000000$
- Exceso 128: el 128 se representará como el 256 ($128 + 128$) en binario natural, es decir, 100000000. El -128 como el 0 ($-128 + 128$) en binario natural, esto es 000000000. Ambos se deben expresar con 9 bits, según el enunciado de la pregunta.
- Complemento a 2: Si se utiliza una representación de 8 bits, el 128 pasa a ser un número negativo –su bit más significativo es 1–, así que tendremos que utilizar 9 bits. El 128 queda como 010000000 y el -128 como 110000000.

■ Pregunta 4

Inicialmente la salida vale 1, pues el biestable almacena un valor $Q = 0$. Esto lleva los valores 10 a las entradas JK, con lo que al siguiente ciclo de reloj, el valor almacenado en el biestable cambiará a 1, y la salida a 0. Pero ahora tendremos 01 en las entradas JK, por lo que con el siguiente ciclo de reloj el valor de Q volverá a ser cero, y así sucesivamente. En definitiva, la salida del circuito será 1, 0, 1, 0, 1 ...

■ Pregunta 5

Es sencillo ver que la salida del circuito siempre vale 0.

2. Problemas

■ Problema 1

El código mostrado es similar a la respuesta a un problema planteado en clase. El valor de $d1$ determina cuantos 1 tendrá $d2$ a la salida. En este caso, como la comparación del salto es *mayor o igual que cero* el número de 1 en $d2$ será el valor de $d1 + 1$. $d3$ y $d4$ se utilizan para realizar la multiplicación por 10. Efectivamente, en cada iteración $d3$ vale $2 \times d2$ y $d4$ $8 \times d2$, así que su suma da $10 \times d2$. Tras esta explicación es fácil ver que:

- La instrucción de salto se ejecuta 5 veces, 4 saltando al inicio del bucle y la última sin verificarse el salto.
- Los valores de los registros a la salida son $d0 = 0$, $d1 = -1$, $d2 = 11111$, $d3 = 11110$ y $d4 = 8888$.

■ Problema 2

Se deja planteado...

■ Problema 3

Evidentemente el resultado no sería el mismo, pues al solapar las fases tendríamos dos problemas:

- Las instrucciones operan con valores cuando aún no han sido escritos por la instrucción anterior, pues la fase de lectura de operandos de la instrucción segunda se da antes que la de escritura de resultados de la primera.
- Lo mismo ocurre con los saltos: desde que empieza la instrucción de salto hasta que el valor del contador de programa se modifica, se adquieren varias instrucciones que en el modo no solapado no entrarían en la unidad central de proceso.

Otro aspecto menos importante –de acuerdo con lo que se pregunta– es la mayor velocidad de la ejecución solapada.