
Arquitectura de Computadoras

Parcial 2018 – 1/12/2018

- ◆ *El parcial consta de 5 preguntas que se deben responder por escrito.*
- ◆ *Se deben completar TODAS las hojas con el nombre y el número de cédula.*
- ◆ *Las hojas deben numerarse y debe indicarse claramente el total de hojas utilizadas en la primer hoja (incluya el texto TOTAL DE HOJAS: [hojas utilizadas]).*
- ◆ *Se debe utilizar solamente una carilla por hoja y se debe iniciar cada respuesta en una hoja diferente.*
- ◆ *No se puede utilizar material de ningún tipo. Se deben apagar los celulares.*
- ◆ *La duración del parcial es de una hora y media. Se dispone de cartillas Intel 8086.*
- ◆ *Se contestarán preguntas sobre la letra del parcial hasta 20 minutos antes de la hora de finalización.*

Pregunta 1

Compare los estilos de diseño CISC y RISC aplicados en tres de los elementos que caracterizan a una arquitectura Von Neumann.

Respuesta:

Con respecto al set de instrucciones, las arquitecturas CISC suelen tener un conjunto mucho más amplio de instrucciones que las arquitecturas RISC (que como su sigla lo sugiere tienen un conjunto reducido de instrucciones). A nivel de formato de instrucción, las arquitecturas CISC suelen implementar instrucciones de largo variable, mientras que las arquitecturas RISC utilizan una codificación de largo fijo. Por último, una tercer diferencia se puede encontrar en el set de registros, donde las arquitecturas CISC muchas veces ofrecen registros con propósitos específicos, mientras que las RISC disponen de registros uniformes, 'sin personalidad' y, muchas veces, un número mayor.

Pregunta 2

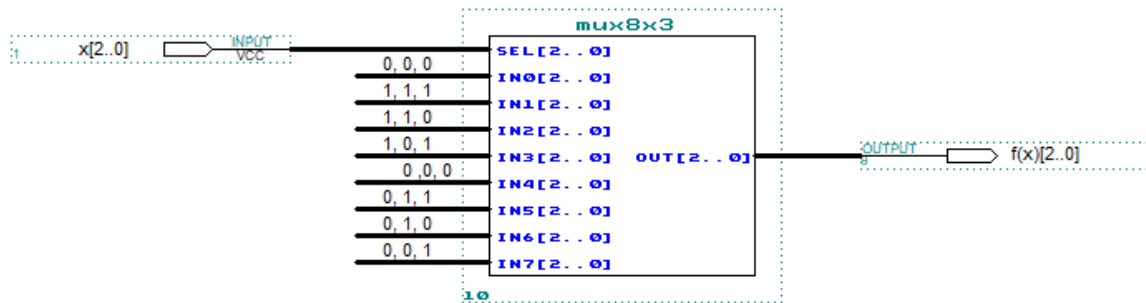
Sea x un número entero con signo representado en complemento a dos de 3 bits. Implemente la función $f(x) = -x$ (siendo $-3 \leq x \leq 3$) utilizando únicamente multiplexores.

Respuesta:

$f(0) = 0 \Rightarrow f(000) = 000$
 $f(1) = -1 \Rightarrow f(001) = 111$
 $f(2) = -2 \Rightarrow f(010) = 110$
 $f(3) = -3 \Rightarrow f(011) = 101$
 $f(-3) = 3 \Rightarrow f(101) = 011$
 $f(-2) = 2 \Rightarrow f(110) = 010$
 $f(-1) = 1 \Rightarrow f(111) = 001$

Como la función no está definida para -4 (100), en forma arbitraria tomaremos la salida 100 para ese caso.

Esta función se implementará con un mux 8x3. Los tres bits de selección van a tomar como entrada el numero x y los tres bits de salida darán el resultado.



Pregunta 3

En una máquina 8086, en la dirección de memoria de 20 bits 0xA12C7 se encuentra la rutina que atiende la interrupción de identificador 16. Indique las direcciones de memoria relevantes del vector de interrupciones. Especifique el contenido que deben tener las direcciones de memoria para que el vector de interrupciones esté configurado correctamente.

Respuesta:

Las dirección donde se encuentra la rutina se almacena en el vector de interrupciones en 4 bytes. La dirección más baja corresponde al desplazamiento de la rutina, mientras que la más alta corresponde al segmento. Como cada posición del vector ocupa 4 bytes, la interrupción de identificador 16 comienza en la dirección 64 = 0x40. Luego, la dirección absoluta 0xA12C7 se puede escribir como:

0xA000:[0x12C7]

Por lo tanto, en el vector de interrupciones se verá lo siguiente:

Mem[0x40] = C7 (parte baja desplazamiento)
Mem[0x41] = 12 (parte alta desplazamiento)
Mem[0x42] = 00 (parte baja segmento)
Mem[0x43] = A0 (parte alta segmento)

Pregunta 4

Sea una CPU con direcciones de 16bits que direcciona de a byte. Se dispone de una memoria caché de 256 Bytes con tamaño de línea de 4 Bytes. La caché implementa una correspondencia asociativa por conjuntos de 4 vías. Indique (y justifique) cómo se interpreta una dirección de memoria para el sistema de caché indicado. Determine el valor de todos los parámetros relevantes.

Respuesta:

Como la caché tiene 256 bytes y cada línea es de 4 bytes, en total hay $256 / 4 = 64$ líneas en el caché. A su vez como cada conjunto tiene 4 líneas, en total hay $64/4 = 16$ conjuntos, por lo tanto se precisan 4 bits para el campo conjunto. Como la caché tiene líneas de 4 bytes, se precisan 2 bits para el campo byte. Finalmente, el resto de los bits (10) son para el campo tag.

TAG	CONJUNTO	BYTE
15 - - - - -	6 5 - - - -	2 1 - - - 0

Pregunta 5

Describa la función de la entrada CE (Clock Enable) de un flip flop. Justifique por qué se prefiere usar la entrada CE y no usar lógica combinatoria en la entrada de reloj para realizar la función descrita.

Respuesta:

La función de la entrada Clock Enable de un flip-flop es habilitar la señal de reloj hacia dentro del flip flop, por lo que resulta en la habilitación de almacenar nuevos valores en el mismo.

Se prefiere usar la entrada CE y no usar lógica combinatoria para habilitar la señal de reloj debido a que una solución con lógica combinatoria --por ejemplo, mediante un AND a la entrada de CLK del flip-flop entre el reloj del sistema y una señal de activación-- podría generar problemas de falsos pulsos de reloj aplicados al flip-flop, en caso de que la señal de activación no esté perfectamente sincronizada con el reloj general.