Arquitectura de Computadoras

Parcial 2019 – 30/11/2019

- *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

Explique qué son los registros visibles y los invisibles. Mencione cinco registros visibles del procesador 8086.

Respuesta 1:
Los registros visibles son los registros disponibles para uso del programador, que pueden ser utilizados como operandos en las instrucciones.
Los registros invisibles son aquellos registros auxiliares del CPU que son utilizados para los procesos internos, y que no tienen que ser conocidos por el programador.

Cinco registros visibles del procesador 8086: AX, BX, CX, DX, SI.

Pregunta 2

Explique qué es un hazard Read after Write e indique al menos dos técnicas con las que puede mitigarse.

Respuesta 2:
Un hazard Read after Write es un tipo de problema que ocurre en un procesador con pipeline, cuando una instrucción posterior en el orden lógico del programa lee un operando antes de que una instrucción anterior haya podido actualizar su valor.

Ejemplo para el pipeline MIPS:

ADDI **R01**, 0x1000
SUBU R02, **R01**, R03

En el curso se vieron tres posibles maneras de mitigarlo:

- Detectar el hazard y detener el pipeline hasta que se escriba el resultado.

- Que el compilador inserte suficientes instrucciones NOP entre ambas instrucciones, de forma que al leer el operando origen, la instrucción anterior ya haya escrito el resultado.
- Implementar la técnica de *register fowarding*, la cual consiste en agregar conexiones entre las etapas del pipeline para poner a disposición el resultado sin necesidad de que este sea escrito en el registro correspondiente.

Pregunta 3

Describa los pasos que realiza una CPU 8086 al recibir un pedido de interrupción.

Respuesta 3:

Un microprocesador 8086, al recibir un pedido de atención a la interrupción, realiza los siguientes pasos (en caso que las interrupciones estén habilitadas):

- Termina de ejecutar la instrucción actual (en realidad verifica la existencia del pedido al final de un ciclo de instrucción y antes de comenzar el siguiente).
- Obtiene el identificador de la interrupción suministrado por el dispositivo/controlador de interrupciones en el bus de datos, mediante la activación la señal INTA y realizando una lectura (el dispositivo/controlador de interrupciones coloca el identificador en el bus de datos en respuesta al INTA).
- Preserva las banderas del procesador y la dirección de la próxima instrucción a ejecutarse, salvando en el stack el valor actual de los registros FLAGS, CS e IP (en ese orden).
- Enmascara las interrupciones
- Accede al vector de interrupciones utilizando el identificador como indice (realiza accesos a memoria, en las direcciones id_interrupt * 4 y id_interrupt * 4 + 2) para obtener el desplazamiento y segmento de la dirección de la primera instrucción de la rutina de servicio de la interrupción.
- Realiza un *jump far* a la dirección obtenida en el vector de interrupciones.

Pregunta 4

a) Indique qué función cumplen las entradas Chip Select y Output Enable en una ROM.

b) Construya una ROM de 32x8 a partir de una ROM de 16x16.

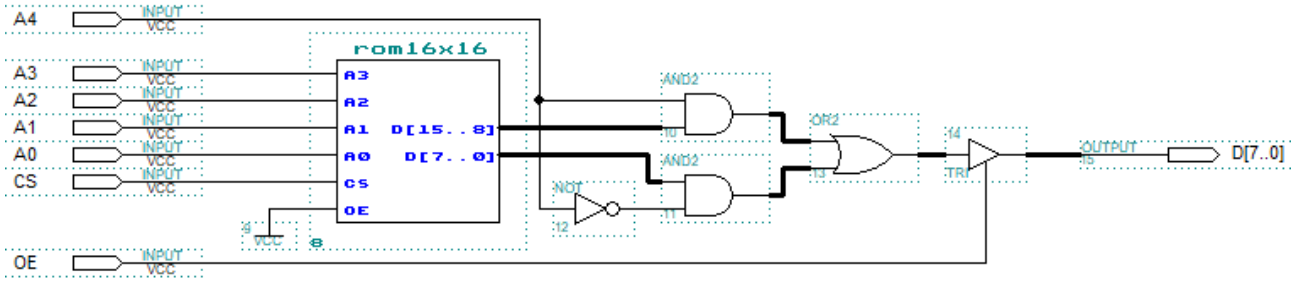
Respuesta 4:

a)

Output Enable (OE): la entrada Output enable tiene como cometido habilitar o no la salidas de una ROM, desde el punto de vista eléctrico. Opera de la siguiente manera: cuando dicha entrada de control está en 0, la salida pasa al "estado de alta impedancia" o “tercer estado” (no influye en el circuito) y cuando la entrada de control está en 1, la salida está en estado lógico 0 ó 1. Esto nos permite realizar ORs “cableados” (sin necesidad de utilizar compuertas).

Chip Select (CS): cuando OE=1, si CS = 0 todas las salidas de la ROM están en 0, con independencia de las entradas de dirección y del "valor" almacenado en la "posición" de la ROM indicada por dicha dirección y si CS = 1 las salidas presentan el contenido de la ROM en la posición señalada por la dirección. Esto nos permite ahorrar en la implementación la estructura de ANDs que deberíamos colocar a la salida de las ROMs a la hora de trabajar con varias ROMs y elegir la salida de la ROM general.

b)



Pregunta 5

a) Defina un circuito multiplexor.
b) Dibuje el circuito interno de un multiplexor con una entrada de control.

Respuesta 5:

a) Un multiplexor es un circuito con “n” entradas de control y 2^n entradas de datos y una salida tal que el valor de la salida coincide con el valor de la entrada de datos señalada por el número binario formado por las entradas de control.

b)