

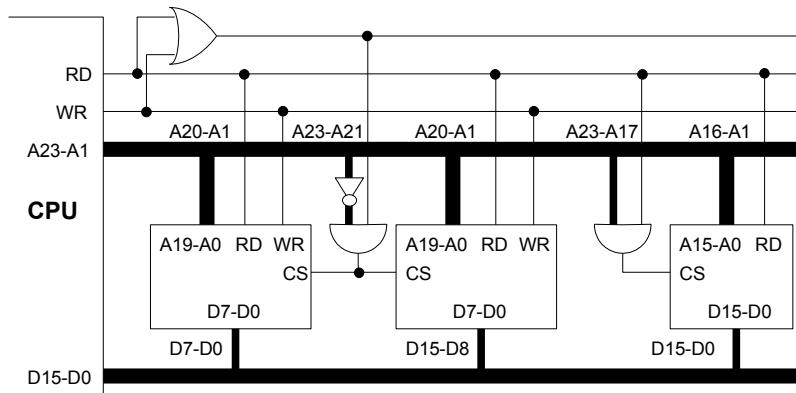
CC41C Introducción al Hardware
Control 2 - Otoño 2007
Profesor: Luis Mateu

Pregunta 1

Parte a.- El siguiente es un programa en assembler x86. Escriba el programa equivalente en C sin usar las instrucción **goto** de C. Preocúpese de reproducir en C el o los *bugs* que contenga el programa en assembler.

<pre>.text .globl unknown unknown: pushl %ebp movl %esp, %ebp movl 8(%ebp), %eax movl 12(%ebp), %edx cmpl \$0, %eax je .L2 cmpl (%eax), %edx je .L2</pre>	<pre>.L7: movl 8(%eax), %eax cmpl \$0, %eax je .L2 cmpl %edx, (%eax) jne .L7 .L2: movl 4(%eax), %eax popl %ebp ret</pre>
---	--

Parte b.- La siguiente figura muestra un procesador y su memoria:



Conteste las siguientes preguntas:

- i. ¿De qué tamaño es la RAM y en qué rango de direcciones se ubica?
- ii. ¿De qué tamaño es la ROM y en qué rango de direcciones se ubica?
- iii. ¿Cuanta es la máxima cantidad de memoria que puede direccionar el procesador?
- iv. ¿Por qué el procesador no tiene la línea de dirección A0?
- v. ¿Después de encenderse el procesador, qué puede decir acerca de la ubicación de la primera dirección de memoria en ser leída?

Pregunta 2

Se desea agregar la instrucción SWAP a M32:

Notación Assembler	Formato instrucción	Significado
SWAP [reg1 + imm], regd		aux= Mem[reg1+{imm}reg2] Mem[reg1+{imm}reg2]= regd
SWAP [reg1 + reg2], regd		regd= aux

La instrucción SWAP intercambia el valor de un registro con el de una palabra en memoria.

- (a) Explique por qué no es posible implementar esta instrucción con el actual diseño de M32.
- (b) Modifique ligeramente M32 de tal forma que sí se pueda implementar SWAP, especificando componentes y señales de control adicionales (redibuje sólo la parte de M32 que cambia).
- (c) Especifique ciclo por ciclo las señales de control que son necesarias para ejecutar SWAP (no es necesario que especifique las señales para la carga de la instrucción y la decodificación).

Observación: La señal de control RD-DEST (de R-SEL) controla si el registro *regd* se va a leer (caso 1) o escribir (caso 0).