

# CC4301 Arquitectura de Computadores

## Solución Control 2

Profesor: Pablo Guerrero P.  
Prof. Aux.: Gaspar Pizarro V.  
Ayudante: Ian Yon Y.

18 de noviembre de 2012

### 1. Conceptos

- a) Palabra:  $w = 0011\ 0111\ 1100\ 0101$   
I  $[\text{Trunc}^{16 \rightarrow 8}]_u \Rightarrow 1100\ 0101$   
II  $[\text{Ext}^{8 \rightarrow 16}]_s \Rightarrow 1111\ 1111\ 1100\ 0101$
- b) I Calcular  $\text{ebx} - \text{eax}$  y almacenarlo en  $\text{ebx}$   
II Negar los bits de la palabra apuntada por  $\text{ebx}$   
III Copia el valor de la palabra ubicada en  $\text{ebp} + \text{eax} * 3 - 12$  a  $\text{edx}$   
IV Copia el valor de  $\text{eax}$  a la dirección apuntada por  $\text{ebp} - 16$
- c) Queda FFFFFFFAD6.
- d) Es un espacio de memoria que funciona con comportamiento LIFO (Last in first out). Esta pila reside en memoria, en la RAM, y es administrada por el procesador, mediante los registros  $\%esp$  y  $\%ebp$ . Sirve para almacenar variables locales e información que no cabe en los registros de la CPU.
- e) I El bus de datos es para enviar datos entre memoria y CPU, en cualquier dirección. Es escrito y leído por los dos.  
II El bus de direcciones es para comunicación unidireccional entre CPU y memoria. Es escrito solamente por la CPU y leído solamente por la memoria.  
III Es para informar la lectura de la memoria. Es escrito por la CPU y leído por la memoria.  
IV Es para informar la escritura de la memoria. Es escrito por la CPU y leído por la memoria.  
V Es para informar a la CPU que la memoria está procesando. Es escrito por la memoria y leído por la CPU.

## 2. Assembler

Paso 1 y paso 9 están fuera de la función. La única variable local half se guardó en un registro, de forma que no fue necesario el paso 4.

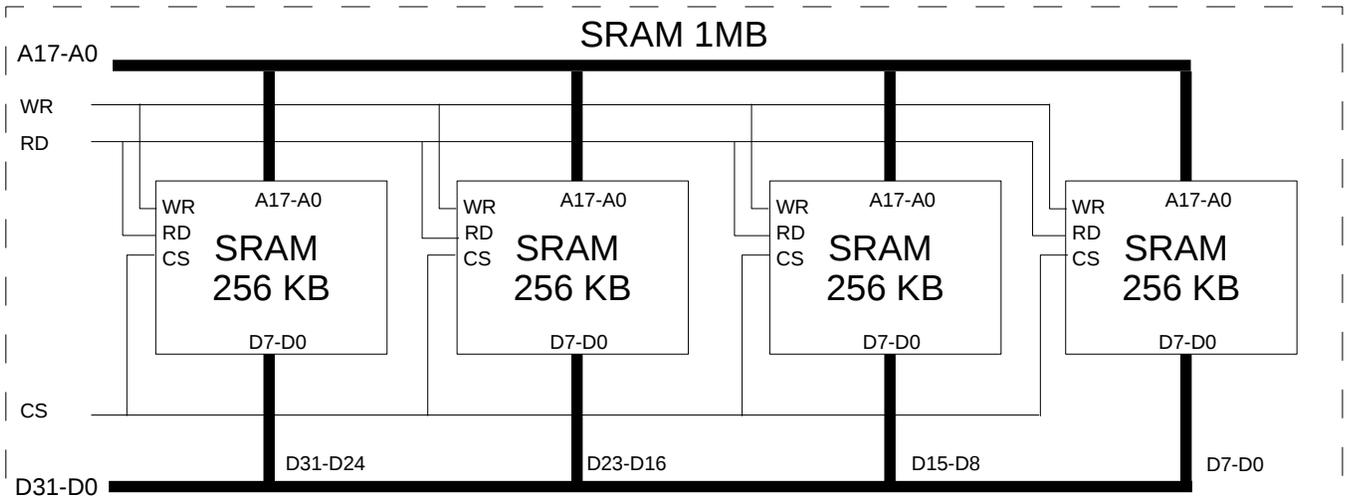
```
.globl _find
_find:
    pushl %ebp          #paso 2
    movl %esp, %ebp    #paso 2
    pushl %esi         #paso 3
    pushl %edi         #paso 3
    pushl %ebx         #paso 3
    movl 8(%ebp), %edx
    movl 12(%ebp), %esi
    movl 16(%ebp), %ecx
    shrl $1, %ecx #paso 5 (en adelante)
    cmpl (%esi, %ecx, 4), %edx
    je end
    pushl %ecx
    cmpl (%esi, %ecx, 4), %edx
    jl mayor

    leal (%esi, %ecx, 4), %ebx
    pushl %ebx
    jmp else
mayor:
    pushl %esi
else:
    pushl %edx
    call _find
end:
    leal (%esi, %ecx, 4), %eax #paso 6
    pushl %ebx #paso 7
    pushl %edi
    pushl %esi
    leave
    ret #paso 8
```

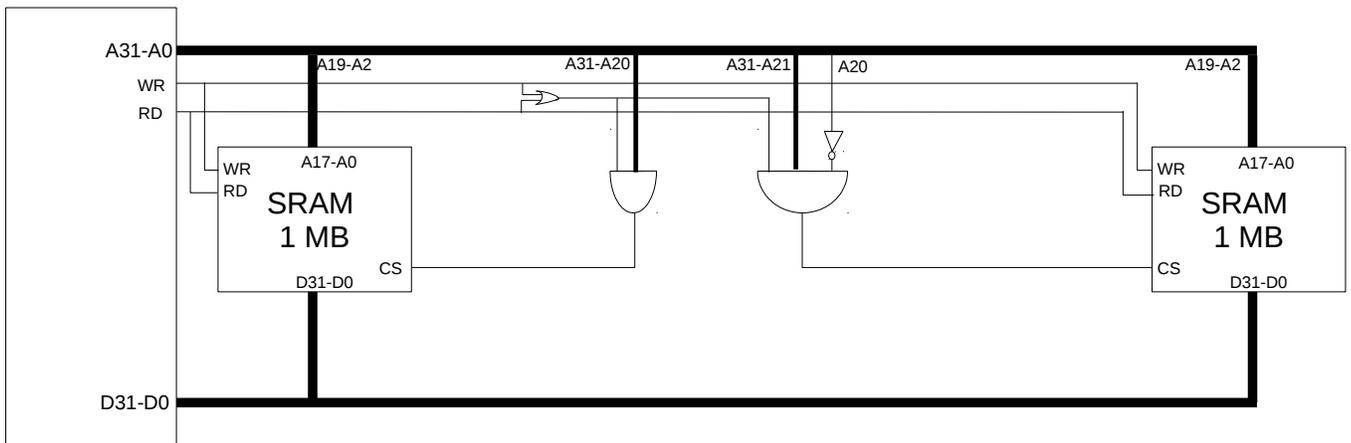
### Asignación de Puntaje

- 0,2 por el cálculo y uso de variable half.
- 0,3 por obtención correcta de argumentos.
- 0,5 por llamada a función.
- 0,5 por correcta bifurcación.
- 0,1 por cada paso identificado correctamente.

### 3. Memoria



(a) Memoria SRAM de 1 MB hecha con los bloques disponibles



(b) Conexión de memorias de 1 MB a CPU

Figura 1: Solución problema 3

#### Asignación de Puntaje

- 0,5 por poner 2 MB.
- 0,5 por completar el bus de datos.
- 0,5 por chip select.
- 0,5 por alineamiento de memoria.