

## Examen Test de Teoría (3.0p)

Todas las preguntas son de elección simple sobre 4 alternativas.

Cada respuesta vale 3/30 si es correcta, 0 si está en blanco o claramente tachada, -1/30 si es errónea.

Anotar las respuestas (a, b, c o d) en la siguiente tabla.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
a	c	d	c	a	b	c	c	d	d	a	c	d	c	b	b	c	d	b	c	a	c	c	a	d	b	a	b	d	d

## Examen Test de Prácticas (4.0p)

Todas las preguntas son de elección simple sobre 4 alternativas.

Cada respuesta vale 4/20 si es correcta, 0 si está en blanco o claramente tachada, -1.33.../20 si es errónea.

Anotar las respuestas (a, b, c o d) en la siguiente tabla.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
a	c	d	c	b	c	b	d	a	c	d	d	d	b	b	c	b	d	b	d

Errata detectada  
por E.Gómez,  
ver Sept'2013

## Examen de Problemas (3.0p)

### 1. Ensamblador (0.5 puntos).

```
max:
    pushl %ebp
    movl %esp, %ebp
    movl 8(%ebp), %eax
    cmpl 12(%ebp), %eax
    jle .else
    movl 8(%ebp), %eax
    jmp .cont
.else:
    movl 12(%ebp), %eax
.cont:
    popl %ebp
    ret
```

```
max:
    pushl %ebp
    movl %esp, %ebp
    movl 8(%ebp), %edx
    movl 12(%ebp), %eax
    cmpl %edx, %eax
    jge .cont
.else:
    #innecesario
    movl %edx, %eax
.cont:
    popl %ebp
    ret
```

### 2. Ensamblador (0.5 puntos).

ASM	función C
A	guru
B	rookie
C	hacker

Ver hojas adjuntas para

- Unidad de control (0.5 puntos).
- Entrada/Salida (0.5 puntos).
- Jerarquía de memoria (0.25 puntos).
- Diseño del sistema de memoria (0.5 puntos).
- Memoria cache (0.25 puntos).

## ① Ensamblador

```

int max (int a, int b)
{
    if (a > b)
        return a;
    else
        return b;
}

```



Alternativa 1

```

pushl %ebp
movl %esp, %ebp
movl 8(%ebp), %eax
cmpl 12(%ebp), %eax
jle .L0
movl 12(%ebp), %eax
.L0: popl %ebp
ret

```

Alternativa 2

```

pushl %ebp
movl %esp, %ebp
movl 8(%ebp), %eax
movl 12(%ebp), %edx # caller save
cmpl %edx, %eax
jle .L0
movl %edx, %eax
.L0: popl %ebp
ret

```

8 instrucciones,  
2 o 3 accesos  
a memoria  
(aparte de  
pushl/popl)

9 instrucciones  
2 accesos a  
memoria

## ② Ensamblador

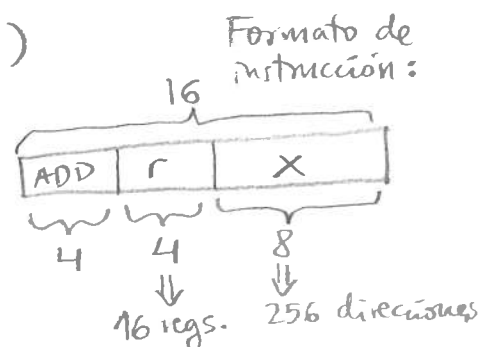
- A. guru
- B. rookie
- C. hacker

(el único que devuelve una dirección)

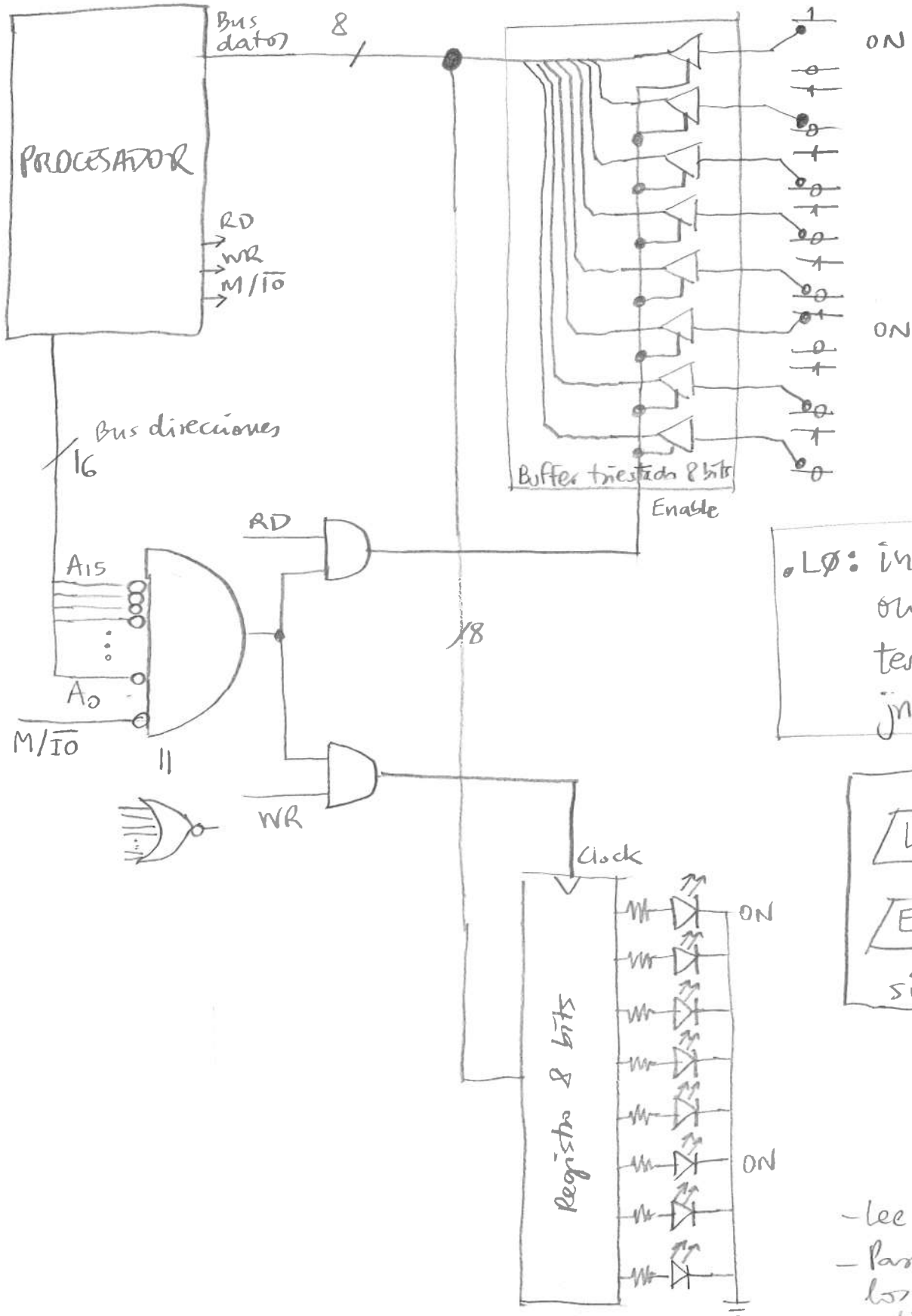
## ③ Unidad de control

Fetch:  $MAR \leftarrow PC[7:0]$ ;  $A \leftarrow PC$   
 $IR \leftarrow M[MAR]$   
 $PC \leftarrow A + 1$ ; goto  $\{ (IR) \}$

ADD:  $A \leftarrow r$   
 $MAR \leftarrow IR[7:0]$   
 $B \leftarrow M[MAR]$   
 $A \leftarrow A + B$   
 $M[MAR] \leftarrow A$ ; goto Fetch

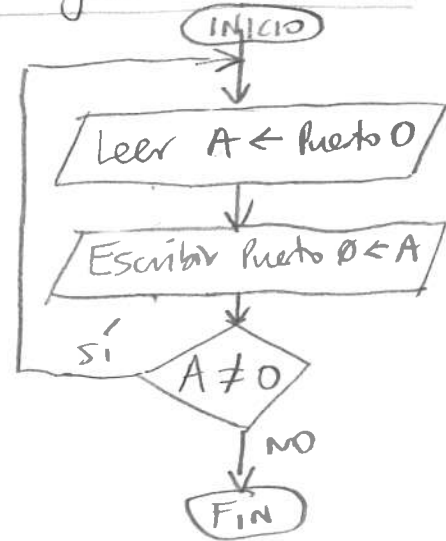


# ④ Entrada / salida



Leer mientras  
algun conmutador  
esté en ON

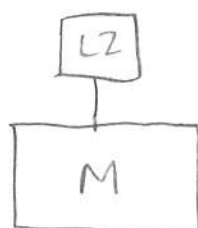
.LØ: in Ø, %al  
out %al, Ø  
test %al, %al  
jnz .LØ



- lee y escribe
- Para cuando  
los conmutadores  
estén a OFF

# 5) Jerarquía de memoria

Opción a

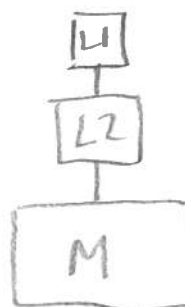


$$A_1 = 0 \quad F_0 = 1$$

$$t_1 = 1,9 \text{ ns} \\ A_1 = 0,98 \quad F_1 = 0,02$$

$$t_2 = 55 \text{ ns} \\ A_2 = 1 \quad F_2 = 0$$

Opción b



$$A_1 = 0 \quad F_0 = 1$$

$$t_1 = 0,6 \text{ ns} \\ A_1 = 0,95 \quad F_1 = 0,05$$

$$t_2 = 1,9 \text{ ns} \\ A_2 = 0,98 \quad F_2 = 0,02$$

$$t_3 = 55 \text{ ns} \\ A_3 = 1 \quad F_3 = 0$$

$$\begin{aligned} \overline{T} &= \sum_{i=1}^2 F_{i-1} t_i = t_1 + F_1 t_2 = \\ &= 1,9 \text{ ns} + 0,02 \cdot 55 \text{ ns} = \\ &= 1,9 \text{ ns} + 1,1 \text{ ns} = 3 \text{ ns} \end{aligned}$$

$$\begin{aligned} \overline{T} &= \sum_{i=1}^3 F_{i-1} t_i = t_1 + F_1 t_2 + F_2 t_3 = \\ &= 0,6 \text{ ns} + 0,05 \cdot 1,9 \text{ ns} + 0,02 \cdot 55 \text{ ns} = \\ &= 0,6 \text{ ns} + 0,095 \text{ ns} + 1,1 \text{ ns} = 1,795 \text{ ns} \end{aligned}$$

$$\begin{aligned} \overline{T} &= \sum_{i=1}^2 a_i T_i = a_1 t_1 + a_2 (t_1 + t_2) = \\ &= 0,98 \cdot 1,9 \text{ ns} + 0,02 (1,9 \text{ ns} + 55 \text{ ns}) = \\ &= 1,862 \text{ ns} + 1,138 \text{ ns} = 3 \text{ ns} \end{aligned}$$

$$\begin{aligned} \overline{T} &= \sum_{i=1}^3 a_i T_i = a_1 t_1 + a_2 (t_1 + t_2) + a_3 (t_1 + t_2 + t_3) = \\ &= 0,95 \cdot 0,6 \text{ ns} + 0,03 (0,6 \text{ ns} + 1,9 \text{ ns}) + \\ &\quad + 0,02 (0,6 \text{ ns} + 1,9 \text{ ns} + 55 \text{ ns}) = \\ &= 0,57 \text{ ns} + 0,075 \text{ ns} + 1,15 \text{ ns} = 1,795 \text{ ns} \end{aligned}$$

$$a_1 = A_1 = 0,98$$

$$a_2 = A_2 - A_1 = 1 - 0,98 = 0,02$$

$$a_1 = A_1 = 0,95$$

$$a_2 = A_2 - A_1 = 0,98 - 0,95 = 0,03$$

$$a_3 = A_3 - A_2 = 1 - 0,98 = 0,02$$

$$\text{Mejora} = \frac{3 \text{ ns}}{1,795 \text{ ns}} = 1,6713 \dots$$

$$\text{Mejora} = \frac{V_b}{V_a} = \frac{\frac{x}{t_b}}{\frac{x}{t_a}} = \frac{t_a}{t_b} = 1,6713 \dots \Rightarrow V_b = 1,6713 V_a$$

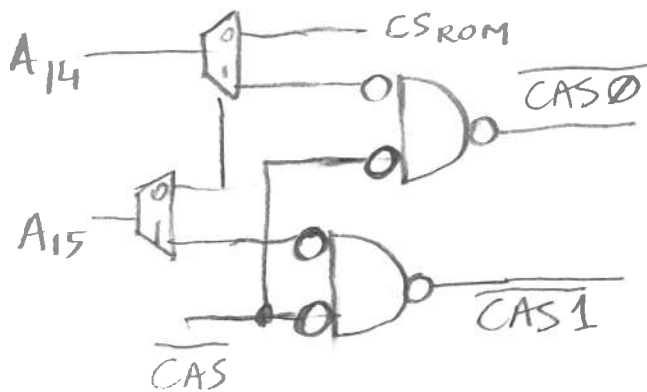
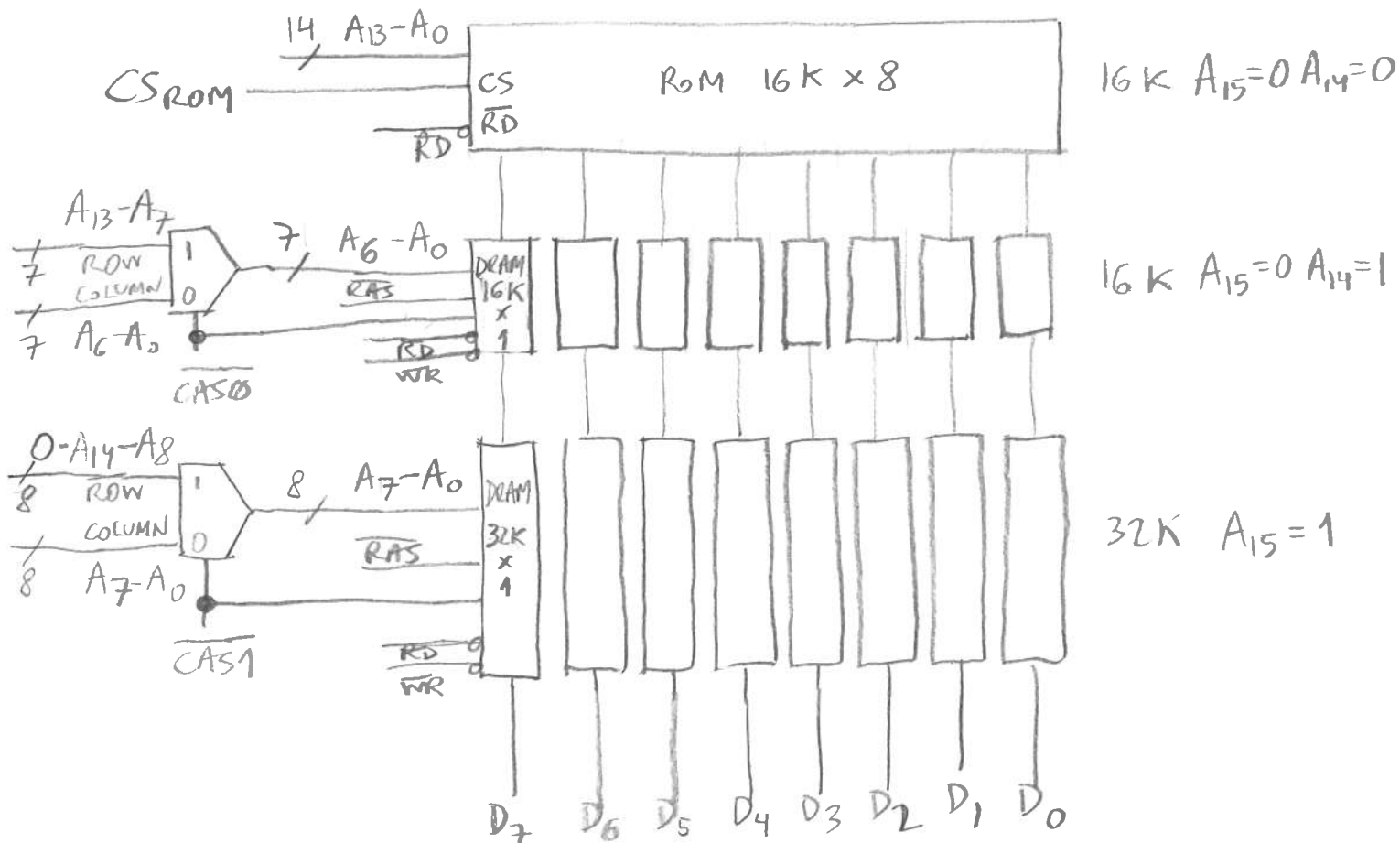
$V_b$  es un 67,13% mejor que  $V_a$



# ⑥ Diseño del sistema de memoria

Mapa de memoria:

ROM	0x0000
	0x3FFF
DRAM	0x4000
	0x7FFF
DRAM	0x8000
	...
	0xFFFF

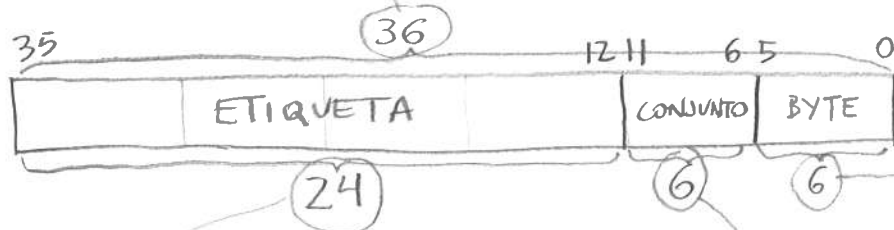
Esquema:



 =   
 NAND con entradas negadas = OR

## 7) Memoria cache

$$\begin{aligned} 64 \text{ GB} &= 2^{36} \text{ B} \\ 32 \text{ KB} &= 2^{15} \text{ B} \\ 64 \text{ B} &= 2^6 \text{ B} \\ 8 \text{ vías} &= 2^3 \text{ líneas/conj} \end{aligned}$$



$$\frac{2^{15} \text{ B}}{2^6 \frac{\text{B}}{\text{línea}}} = 2^9 \text{ líneas en caché}$$

$$\frac{2^9 \text{ líneas}}{2^3 \frac{\text{líneas}}{\text{conj.}}} = 2^6 \text{ conjuntos}$$

$$\frac{2^{36} \text{ B}}{2^6 \frac{\text{B}}{\text{línea}}} = 2^{30} \text{ líneas en M.P.}$$

$$\frac{2^{30} \text{ líneas en M.P.}}{2^6 \text{ conjuntos}} = 2^{24} \frac{\text{líneas distintas M.P.}}{\text{conjunto}}$$

1. Procesador direcciona las 8 memorias de etiquetas con los 6 bits  $A_{11}-A_6$ , leyendo 8 etiquetas de 24 bits
2. Se comparan las 8 etiquetas leídas con los bits  $A_{35}-A_{12}$ . Si coinciden  $\Rightarrow$  acierto
3. Si acierto  $\Rightarrow$  se lee el byte indicado por  $A_{11}-A_0$  de la memoria de datos correspondiente al acierto (una de las 8 memorias de datos, la de la vía que ha dado acierto).