

Se tiene un procesador genérico (no es Intel) con 32 líneas de BA y 16 líneas de BD. Un programador necesita memoria suficiente para poder ubicar un programa de 1GB pero también quiere que la RAM o las RAMs en la que esté ubicado dicho programa tenga la menor cantidad de direcciones posibles para que quepa dicho programa. Es decir que al ubicar el programa deben sobrar la menor cantidad de direcciones posibles en la RAM o las RAMs. Además nos dice que su programa comienza en la mitad del mapa de memoria. Se cuenta con las siguientes RAMs:

1Gx16

512Mx8

2Gx8

Se tiene libertad para elegir entre cualquiera de ellas y cualquier cantidad. Por último, se solicita ubicar una ROM de 1Gx16 al comienzo del mapa.

Indicar la RAM o las RAMs que habría que elegir y dibujar el mapa de memoria, diagrama y decodificación (como en el parcial). Se puede elegir un único decodificador de entre los siguientes: 2 a 4, 3 a 8 y 4 a 16. No se cuenta con compuertas ni ningún otro tipo de Integrados.

10-50

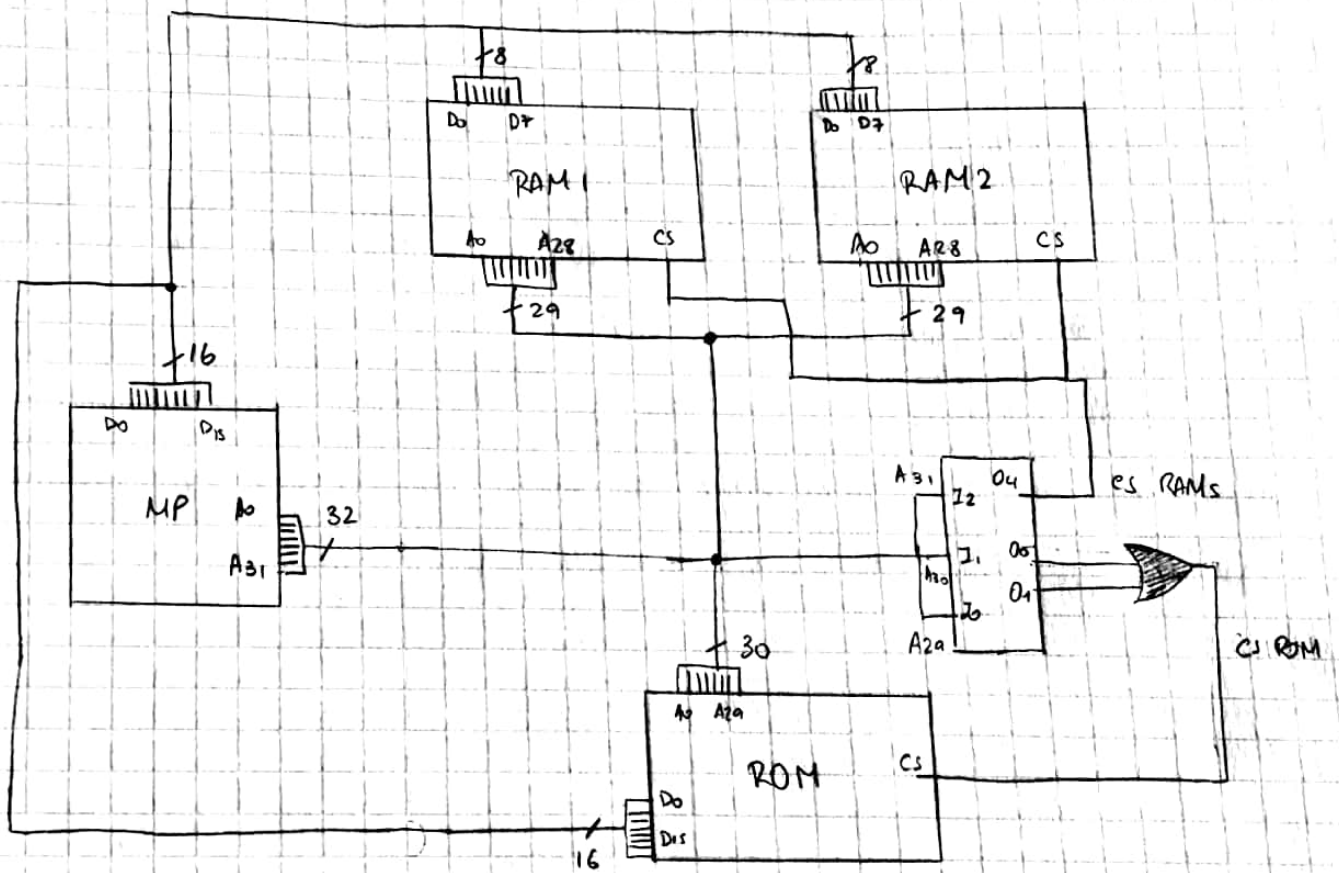


## Stack

Sea la siguiente función escrita en C para un procesador con arquitectura x86:

```
int factorial(int n) {  
    int aux, fact;  
    if (n == 0)  
        return 1;  
    aux = n;  
    fact = factorial(n-1);  
    return fact * aux;  
}
```

Mostrar el estado de la pila si es invocado con  $n = 3$ .



Empieza en 7

HOJA Nº

FECHA

0000h

11 fact:  
aux:

10

8 fact:  
aux:

7

5 fact:  
aux:

4

2 fact:  
aux:  
n

1

EBP POP 12

retorno POP 13

0

1 → 14

1 → 11

EBP POP 15

retorno POP 16

1

1 → 17

2 → 12

EBP POP 18

retorno POP 19

2

2 → 20

3 → 13

EBP POP 21

retorno POP 22

3

1. primer llamado, se pusha EBP y retorno

2. espacio para aux y fact

3. se guarda 3 en aux

4. se pusha m-1, retorno y EBP

5. espacio para aux y fact

6. se guarda 2 en aux

7. se pusha m-1, retorno y EBP

8. espacio para aux y fact

9. se guarda 1 en aux

10. se pusha m-1, retorno y EBP

11. se hace espacio para aux y fact

12. Se popa EBP y se asigna

13. Se popa retorno y se asigna a EBP

14. Se guarda 1 en fact

15. Se popa EBP y se asigna

16. Se popa retorno y se asigna

17. Se guarda 1 en fact

18. Se popa EBP y se asigna

19. se popa retorno y se asigna

20. Se guarda 2 en fact

21. Se popa EBP y se asigna

22. Se popa retorno y se asigna

el segundo llamado retorno 2, por lo cual el primer llamado retorna

$$3 \times 2 = 6$$

NOTA