

Ejercicio 1:

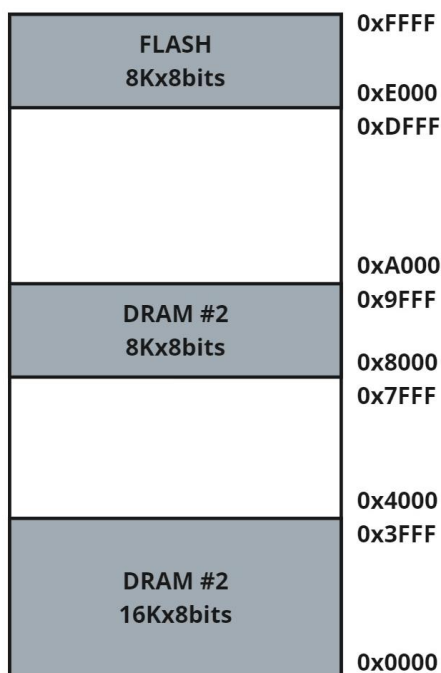
Dadas las siguientes sentencias en assembler LEGv8

```
LSL X9, X0, #3
ADD X9, X9, X8
LDUR X10, [X9, #0]
ADDI X9, X9, #8
LDUR X11, [X9, #0]
ADD X9, X10, X11
STUR X9, [X8, #128]
```

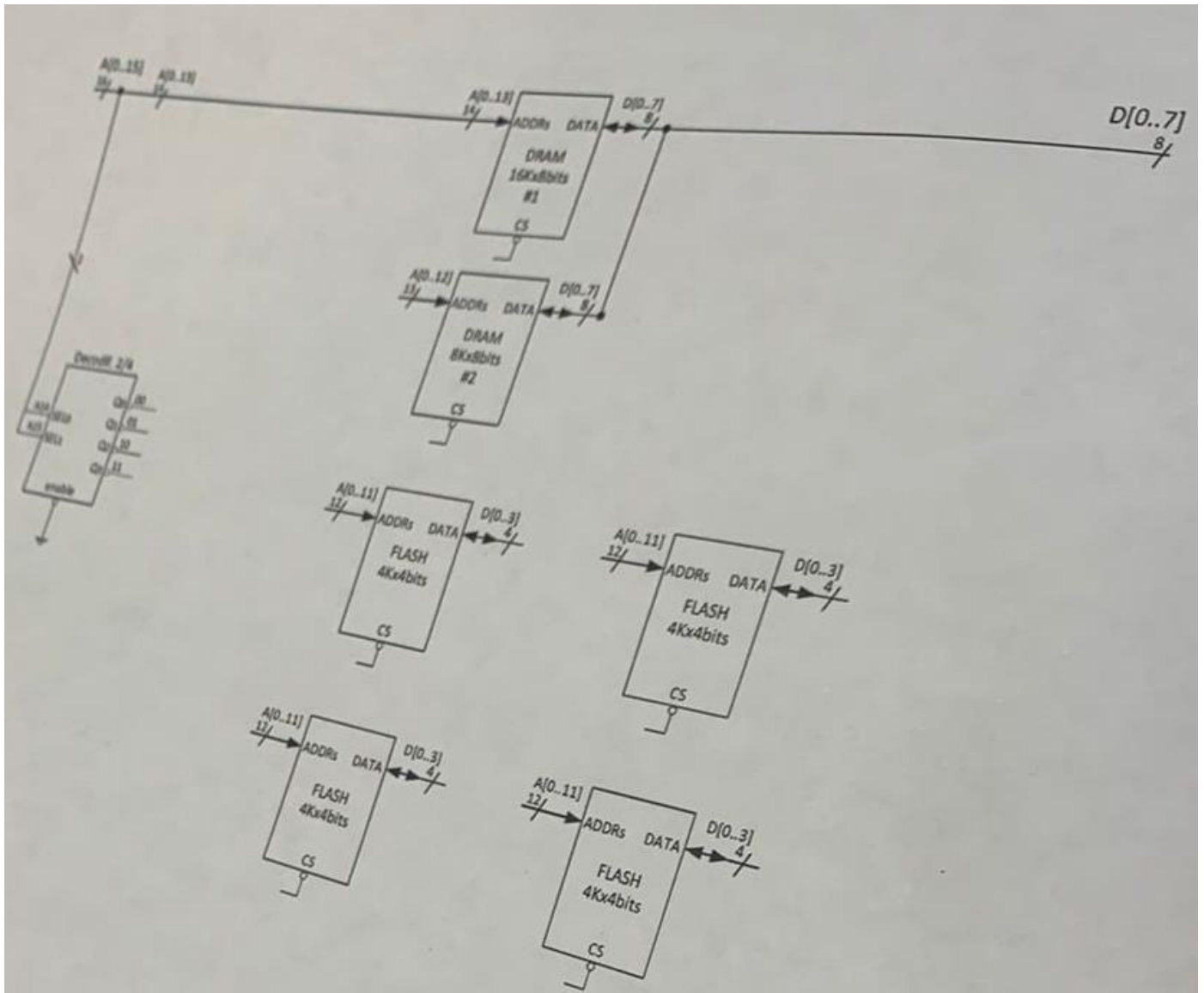
Escribir la secuencia mínima de código “C” asumiendo que los registros X0 y X8 contienen la variable “i” y la dirección base del arreglo “A” respectivamente.

Ejercicio 2:

Para un procesador de 8 bits de datos, el cual es capaz de direccionar 64K palabras de memoria principal, se desea implementar un sistema de memoria como el mostrado en el mapa de la figura.



- Completar las conexiones necesarias de la lógica de decodificación del esquema dado a continuación, especificando para cada línea el tamaño y nombre de la/las señales involucradas. De considerarse necesario, es posible agregar decodificadores del tamaño requerido.
- Indicar si la implementación realizada genera posiciones imagen o espejo. De ser así, especificar en el mapa la aplicación de las mismas.



Ejercicio 3:

Considerando que un sistema tiene una CACHE para DATOS de correspondencia ASOCIATIVA POR CONJUNTOS de 2 VIAS de 2Mbits y 4 palabras por línea, sobre un procesador de 32 bits, CPI = 1, que resuelve todos los data y control de hazard sin necesidad de stalls, y que tiene una memoria principal de 4G palabras de 1 byte cada una.

a) Mostrar el formato de memoria principal:

Memory Address			
Tag	Set	Word	Offset byte

Llenar con 0 en el caso que no corresponda.