

Dado el siguiente código escrito en C:

```
int examen(short a, char b, int v[10], short M[4][4]) {
    int ii;
    short aa;
    short *matriz;
    int *vector;
    ...
    return v[ii];
}
```

- c) **Dibuja** el bloque de activación de la rutina examen, indicando claramente los desplazamientos respecto a **%ebp** y el tamaño de todos los campos.

	-----	ii<- ebp-16
4	ii	
	-----	aa<- ebp-12
4	-- aa	
	-----	*matriz<- ebp-8
4	*matriz	
	-----	*vector<- ebp-4
4	*vector	
	-----	<- ebp
4	ebp old	

4	ret	
	-----	<- ebp+8
4	-- -- a	
	-----	<- ebp+12
4	-- -- -- b	
	-----	<- ebp+16
4	@v	
	-----	<- ebp+20
4	@M	
	-----	<- ebp+24

- d) **Traduce** a ensamblador del x86 la instrucción `return v[ii]` USANDO EL MÍNIMO NÚMERO DE INSTRUCCIONES y suponiendo que la subrutina ha usado los registros `%eax`, `%ebx`, `%ecx` y `%edx`.

```
movl 16(%ebp), %ecx            ; %ecx <- @V
movl -16(%ebp), %eax          ; %ecx <- ii
movl (%ecx,%eax,4), %eax      ; %eax <- V[ii]
popl %ebx                      ; restaurar registros. Sólo es necesario guardar ebx
movl %ebp, %esp               ; deshacer enlace dinámico
popl %ebp
ret
```

COGNOMS:

[illegible]

NOM:

[illegible]

Problema 2. (5 puntos)

Un sistema de vigilancia analiza imágenes, provenientes de una cámara, cada T milisegundos, por lo que el procesamiento de cada imagen debe realizarse en un tiempo igual o inferior a T . Se ha calculado que un procesador P funcionando a una frecuencia F es suficiente para procesar una imagen en T milisegundos. Un análisis de rendimiento y consumo de energía muestra que la rutina HFR (Human Face Recognition) consume el 60% del tiempo y el 80% de la energía necesarios para procesar una imagen.

Se ha decidido incorporar un acelerador hardware (AH) al sistema. AH realiza el procesamiento HFR vía hardware pero consumiendo la cuarta parte de la energía respecto la ejecución de la rutina en P. El resto del código, que llamaremos PNA (parte no acelerada) se sigue ejecutando en P y por tanto no se beneficia del acelerador.

- a) **Calcula** la ganancia en energía del sistema con acelerador (P+AH) respecto al sistema P.

Ley de amdahl:

$$G = 1/(0,2 + 0,8/4) \rightarrow G = 2,5$$

El acelerador AH, además de consumir menos energía, también realiza el procesamiento HFR en menos tiempo. Sabemos que el tiempo de ejecución de PNA en P es proporcional al tiempo de ciclo. Dado que no hay ningún beneficio en procesar una imagen en menos de T milisegundos, podemos reducir la frecuencia de P de forma que la parte no acelerada tarde más (siempre que el conjunto no supere T milisegundos).

- b) **Calcula** el speedup mínimo que debería producir AH respecto la rutina HFR para que podamos reducir la frecuencia de P a la mitad sin que el tiempo de procesamiento de una imagen sea superior a T.

Ley de amdahl generalizada:

El speedup global es 1 (tenemos que tardar lo mismo que sin acelerador).

La parte no acelerada tarda el doble (mitad de frecuencia)

$$1 = 1/(0,4/0,5 + 0,6/x) \rightarrow x = 3$$

El procesador P funciona a un frecuencia de 1 GHz. La parte no acelerada (PNA) que corre sobre P ejecuta 6 millones de instrucciones, realiza 3 millones de operaciones de punto flotante y tarda 9 millones de ciclos.

- c) **Calcula** el CPI y el tiempo de ejecución en milisegundos (Texec) de PNA en P.

$$\text{CPI} = 9 \times 10^6 \text{ ciclos} / 6 \times 10^6 \text{ instrucciones} = \mathbf{1,5 \text{ ciclos / instrucción}}$$

$$\text{Texe} = \text{Ciclos} / F = 9 \times 10^6 \text{ ciclos} / 1\text{GHz} = \mathbf{9 \text{ ms}}$$

- d) **Calcula** los MIPS y MFLOPS de PNA en dicho sistema.

$$\text{MIPS} = \text{Instrucciones} / \text{Texe} \cdot 10^6 = 6 \times 10^6 \text{ instrucciones} / 9 \times 10^3 = \mathbf{667 \text{ MIPS}}$$

$$\text{MFLOPS} = \text{OpsPF} / \text{Texe} * 10^6 = 3 \times 10^6 \text{ opsPF} / 9 * 10^3 = \mathbf{333 \text{ MFLOPS}}$$

El procesador P dispone de una cache con una política de escritura *Copy Back* y *Write Allocate*. En caso de acierto en la cache el tiempo de acceso es de 1 ciclo. En caso de fallo, hay una penalización de 60 ciclos para reemplazar un bloque NO modificado y de 120 ciclos para reemplazar un bloque modificado.

PNA realiza 8 millones de accesos a memoria y tiene una tasa de fallos (miss) del 10%. Sabemos que el 15% de los accesos son escrituras y que la probabilidad de que un bloque haya sido modificado en cache es del 20%.

e) **Calcula** el tiempo medio de acceso a memoria (T_{ma}) en ciclos de la parte no acelerada (PNA).

$$T_{ma} = T_{sa} + m \cdot (P_m \cdot T_{pfm} + P_n \cdot T_{pfn}) = 1 + 0,10 \cdot (0,20 \cdot 120 + 0,8 \cdot 60) = 8,2 \text{ ciclos/acceso}$$

Leer o escribir un bloque en memoria principal consume 100 nanoJoules (nJ).

f) **Calcula** el consumo total de energía de la memoria principal causada por los fallos de cache.

fallo que reemplaza bloque NO modificado genera 1 acceso a MP (lectura bloque)

fallo que reemplaza bloque modificado genera 2 accesos a MP (escritura bloque + lectura bloque)

$$\text{AccesosMP} = \text{accesos} \cdot m \cdot (P_n \cdot 1 + P_m \cdot 2) = 8 \times 10^6 \cdot 0,1 \cdot (0,8 \cdot 1 + 0,2 \cdot 2) = 0,96 \times 10^6 \text{ accesos a MP}$$

$$E = 0,96 \times 10^6 \cdot 100 \text{ nJ} = 96 \text{ mJoules}$$

El procesador P genera direcciones lógicas de 36 bits y direcciones físicas de 24 bits. La jerarquía completa de memoria está compuesta por un TLB, la memoria cache y la memoria principal. El TLB tiene 4 entradas y es completamente asociativo. La cache tiene un tamaño de 64 Kbytes, líneas de 64 bytes y es 4-asociativa. El tamaño de página del sistema es de 4 KBytes. El TLB se accede antes que la cache, por lo que la cache se indexa con direcciones físicas.

g) **Calcula** el número de líneas, vías y conjuntos que tiene la cache. **Especifica claramente** cómo has realizado los cálculos.

$$64 \cdot 1024 \text{ bytes} / 64 \text{ bytes/linea} = 1024 \text{ líneas}$$

$$4\text{-asociativa} \rightarrow 4 \text{ vías}$$

$$1024 \text{ líneas} / 4 \text{ líneas/conjunto} = 256 \text{ conjuntos}$$

El procesador lanza un acceso a la dirección lógica 0xEFABCD012 y sabemos que el contenido del TLB es:

VPN	PPN
0xFABCD0	0xA00
0xBCD012	0xB01
0xEFABCD	0xC02
0xABCD01	0xD03

h) **Indica** a qué dirección física se accede, en qué conjunto de la cache se encuentra el dato y cuál es la etiqueta guardada en memoria cache. **Justifica** la respuesta.

$$\text{VPN} = 0xEFABCD \rightarrow \text{PPN} = 0xC02 \text{ @Física} = 0xC02012$$

VPN (24 bits)	desplaçament (12 bits)
---------------	------------------------

$$\text{Conjunto} = 0x80 \quad \text{TAG} = 0x300$$

TAG (10 bits)	cjt (8)	byte (6)
---------------	---------	----------