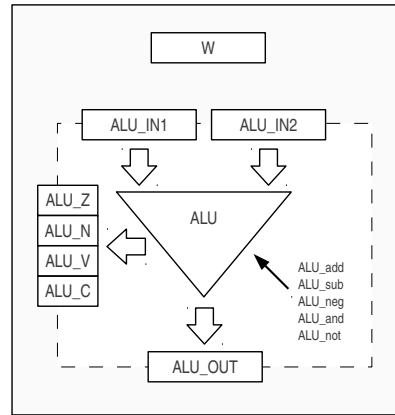


Este examen se aprueba obteniendo al menos **65 puntos**. El parcial **NO** es a libro abierto, pero puede tener la Arquitectura de ORGA1 y dos hojas A4 con apuntes propios. Las soluciones a ejercicios de la práctica que se utilicen deben ser incluidas en el examen.
RECUERDE justificar todas sus respuestas.

Ejercicio 1 *ejercicio1.tex* (25 puntos) Sea un procesador con palabras y direcciones de 16 bits, instrucciones de longitud fija de 32 bits y direccionamiento a *Nibble*¹.

Sabemos que dicho micro, contiene un registro principal de operación W y una ALU que responde a las señales ALU_ADD, ALU_SUB, ALU_AND, ALU_OR y ALU_NOT computando las operaciones correspondientes.

Un esquema incompleto de la microarquitectura se muestra en la figura.



a) Complete la microarquitectura con todo lo que crea necesario para poder soportar las instrucciones detalladas en la tabla.

Agregue nuevos componentes y describa los presentes en la figura. Indique el tamaño de cada registro y de los buses internos.

Justifique la elección de *cada elemento*, teniendo en cuenta que cada uno de ellos debe tener una función específica.

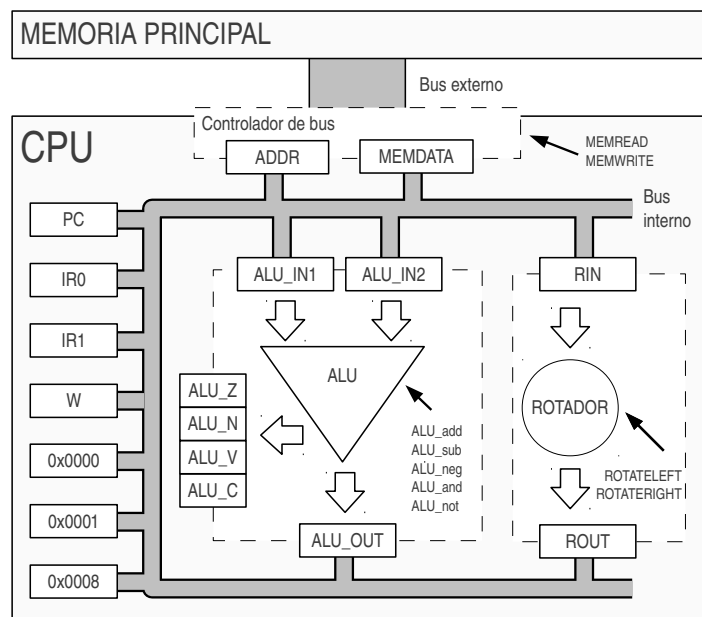
b) Describa cuál es la secuencia de microoperaciones que debe realizar la unidad de control para ejecutar las siguientes instrucciones:

- a) ADDWF
- b) CLRF
- c) DECFSZ
- d) RLF

(no es necesario describir el *fetch*).

Instrucción	Efecto
ADDWF <i>dir</i>	$W \leftarrow W + [dir]$
CLRF <i>dir</i>	$[dir] \leftarrow 0$
CLRW	$W \leftarrow 0$
DECF <i>dir</i>	$[dir] \leftarrow [dir] - 1$
DECFSZ <i>dir</i>	$[dir] \leftarrow [dir] - 1$ e ignora la siguiente instrucción si el resultado es 0
INCF <i>dir</i>	$[dir] \leftarrow [dir] + 1$
INCFSZ <i>dir</i>	$[dir] \leftarrow [dir] + 1$ e ignora la siguiente instrucción si el resultado es 0
MOVF <i>dir</i>	$W \leftarrow [dir]$
MOVWF <i>dir</i>	$[dir] \leftarrow W$
RLF	$W \leftarrow W/2$
RRF	$W \leftarrow W * 2$

Solución.



1.

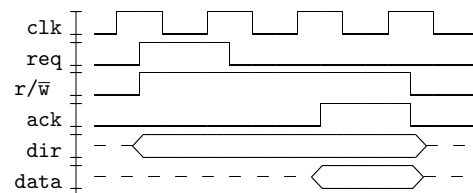
¹Medio *byte*, es decir, cuatro bits

2.
 - ADDWF:
 - ADDR := IR1
 - MEMREAD
 - ALUIN1 := MEMDATA
 - ALUIN2 := W
 - ALUADD
 - W := ALUOUT
 - CLRF:
 - ADDR := IR1
 - MEMDATA := 0x0000
 - MEMWRITE
 - DECFSZ:
 - ADDR := IR1
 - MEMREAD
 - ALUIN1 := MEMDATA
 - ALUIN2 := 0x0001
 - ALUSUB
 - MEMDATA := ALUOUT
 - MEMWRITE
 - IF ALUZ == 1
 - ALUIN1 := PC
 - ALUIN2 := 0x0008
 - ALUADD
 - PC := ALUOUT
 - ENDIF
 - RLF:
 - RIN := W
 - ROTATELEFT
 - W := ROUT

Ejercicio 2 *ejercicio4.tex* (25 puntos) En una computadora que respeta el esquema 1 de la guía de buses (sólo se comunican memoria y CPU), se utiliza un bus síncrono en el cual se producen lecturas como se indica en la figura.

Se desea agregar la posibilidad de realizar lecturas en ráfagas de cuatro palabras.

En ese caso, cuando el CPU realice el pedido de una dirección, la memoria deberá devolver la palabra que comienza en esa dirección y las tres palabras siguientes.

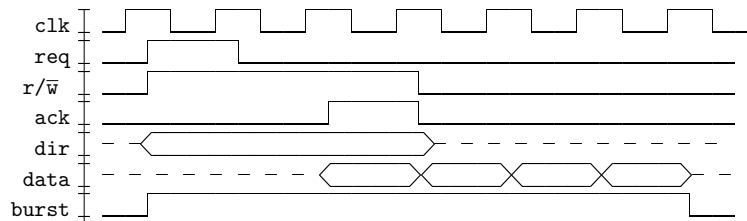


- a) ¿Es necesario cambiar la arquitectura del bus para soportar esta nueva funcionalidad? Explique cómo se brinda dicho soporte teniendo en cuenta su respuesta.
- b) Dibuje un diagrama de tiempo de una lectura en ráfagas. Sea prolijo.
- c) Realice una máquina de estados para la memoria, teniendo en cuenta sólo las acciones de una lectura en ráfagas.

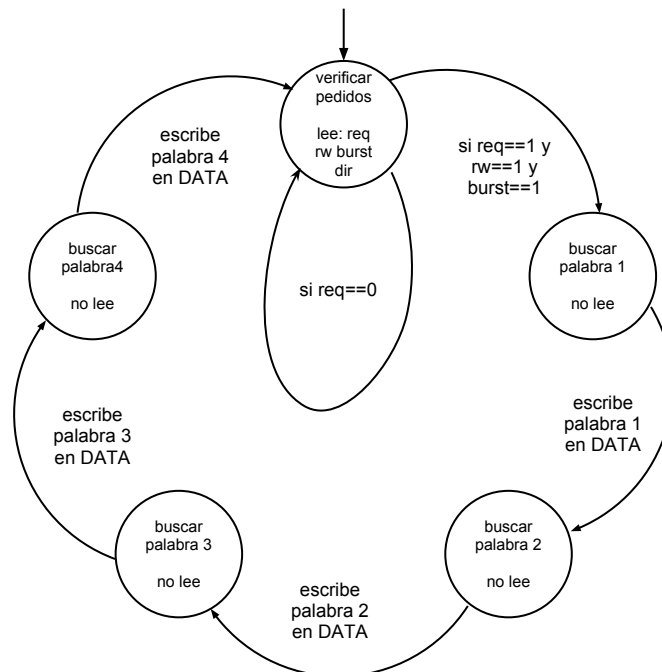
Solución.

Una posible solución es agregar una línea más que permita a la CPU indicar que quiere realizar una lectura en ráfagas. Debería levantarla junto con r/\bar{w} y req y dejarla levantada hasta el último ciclo de la transferencia. Al notar esta señal, la memoria devuelve las cuatro palabras, una por cada ciclo del reloj del bus. El resto del protocolo queda igual.

- Diagrama de tiempo



- Máquina de estados para la memoria



Ejercicio 3 *ejercicio2.tex* (25 puntos) Para evitar futuros paros de camiones y el consecuente desabastecimiento que conllevan, el sindicato de empresarios planeaban adquirir computadoras inteligentes YANOMO. Estas computadoras, fabricadas en China, están diseñadas específicamente para conducir camiones de forma automática. Sin embargo, el elevado costo del producto y el gran número de unidades que se requieren dificultan su compra. Buscando una solución más económica, decidieron comprar una sola YANOMO y encargarle a algunos docentes de esta materia que la repliquen.

Hasta el momento, pudimos determinar que se trata de una arquitectura de 16 bits, que implementa direcciones de memoria de 20 bits y que direcciona *a byte*. Del manual se sabe que cuenta con una caché de 4 KB de capacidad con 32 bytes por línea y una política de reemplazo LRU, pero como GOOGLE TRANSLATOR no es tan exacto, no se pudo determinar el tipo de la caché. Se sospecha que puede ser asociativa por conjuntos de dos vías o de correspondencia directa.

Frente a esto, decidimos recurrir a los alumnos dado que por haber hecho todas las prácticas de la materia tienen un profundo conocimiento de las memorias caché.

- Indique cómo se distribuirían los bits de una dirección de memoria en los campos correspondientes para cada uno de los tipos de cachés mencionadas. Explique cómo se obtiene la ubicación del dato en la caché a partir de una dirección de la memoria principal.
- Buscando en el sitio web del fabricante, hallamos un benchmark para la caché de la YANOMO. Según esa información, al realizar los siguientes pedidos se obtiene un hit-rate mayor a 6/11.

- | | | | |
|----------|----------|----------|-----------|
| 1) CD5E4 | 4) C3DE9 | 7) 5633F | 10) C3DEF |
| 2) CD5E6 | 5) C3DE1 | 8) 56336 | |
| 3) 5633F | 6) CD5E2 | 9) FD5EB | 11) FD5E2 |

Simule estos pedidos ante una caché asociativa por conjuntos de dos vías. Indique claramente los hits, misses, desalojos y accesos desalineados cuando corresponda.

- c) Si el seguimiento no tiene el resultado publicado en el sitio del fabricante, explique si en la caché de correspondencia directa se puede alcanzar o si hay un error en la página web.

Solución.

Info memoria:

- tipo de direccionamiento: A byte
- tam_palabra = 2 bytes
- tam_memoria = tam_unidad_dir * $2^{\text{tam_dir_memoria}}$ = 1 B * 2^{20} = 1 MB

Info caché:

- tam_caché = 4 KB = 2^{12} B
- tam_linea_caché = 32 B = 2^5 B
- cant_lineas_caché = 128 = 2^7
- cant_vías = 2

- a) ■ Correspondencia Directa:

$$\text{cant_bloques_caché} = \frac{\text{tam_memoria}}{\text{tam_bloque}}$$

Como es una caché de *Correspondencia Directa* el tamaño del bloque es igual al de la caché, luego:

$$\text{cant_bloques_caché} = \frac{\text{tam_memoria}}{\text{tam_cache}} = \frac{1 \text{ MB}}{4 \text{ KB}} = \frac{2^{20}}{2^{12}} = 2^8$$

Finalmente, para obtener el tamaño de los campos *tag*, *linea* e *index* aplico \log_2 a los valores de **cant_bloques_caché**, **cant_líneas_bloque** (que es igual a **cant_líneas_caché**) y **tam_línea** respectivamente:

tag	linea	index
8	7	5

- Asociativa por Conjuntos de 2 Vías:

$$\text{cant_bloques_caché} = \frac{\text{tam_memoria}}{\text{tam_bloque}}$$

Como es una caché *Asociativa por Conjuntos de 2 Vías* el tamaño del bloque es igual al de una vía, luego es necesario calcular el tamaño de una vía:

$$\text{cant_líneas_vía} = \frac{\text{cant_lineas_cache}}{\text{cant_vias}} = \frac{128 \text{ linea}}{2 \text{ via}} = 64 \frac{\text{linea}}{\text{via}} = 2^6 \frac{\text{linea}}{\text{via}}$$

$$\text{tam_vía} = \text{cant_líneas_vía} * \text{tam_línea_caché} = 2^6 \frac{\text{linea}}{\text{via}} * 2^5 \frac{\text{B}}{\text{linea}} = 2^{11} \frac{\text{B}}{\text{via}}$$

Entonces, sabiendo el valor de **tam_vía** ahora puedo calcular:

$$\text{cant_bloques_caché} = \frac{\text{tam_memoria}}{\text{tam_bloque}} = \frac{\text{tam_memoria}}{\text{tam_vía}} = \frac{2^{20} \text{ B}}{2^{11} \text{ B}} = 2^9$$

Finalmente, para obtener el tamaño de los campos *tag*, *linea* e *index* aplico \log_2 a los valores de **cant_bloques_caché**, **cant_líneas_bloque** (que es igual a **cant_líneas_vía**) y **tam_línea** respectivamente:

tag	linea	index
9	6	5

- b) Seguimiento de los pedidos a memoria suponiendo una caché de tipo *Asociativa por Conjuntos de 2 Vías*

direccion	tag (9 bits)	linea (6 bits)	index (5 bits)	resultado	estado cache	notas
CD5E4	410d 19Ah	47d 2Fh	4	Miss	{47:410}	cargué 47:410
CD5E6	410d 19Ah	47d 2Fh	6	Hit	{47:410}	
5633F	172d ACh	25d 19h	31d 1Fh	Miss	{26:172}{25:172}{47:410}	acceso desalineado cargué 25:172 y 26:172
C3DE9	391d 187h	47d 2Fh	9	Miss	{47:391}{26:172}{25:172}{47:410}	cargué 47:391
C3DE1	391d 187h	47d 2Fh	1	Hit	{47:391}{26:172}{25:172}{47:410}	
CD5E2	410d 19Ah	47d 2Fh	2	Hit	{47:410}{47:391}{26:172}{25:172}	
5633F	172d ACh	25d 19h	31d 1Fh	Hit	{25:172}{47:410}{47:391}{26:172}	acceso desalineado
56336	172d ACh	25d 19h	22d Bh	Hit	{25:172}{47:410}{47:391}{26:172}	
FD5EB	506d 1FAh	47d 2Fh	11d Bh	Miss	{47:506}{25:172}{47:410}{26:172}	desalojé 47:391 y cargué 47:506
C3DEF	391d 187h	47d 2Fh	15d Fh	Miss	{47:391}{47:506}{25:172}{26:172}	desalojé 47:410 y cargué 47:391
FD5E2	506d 1FAh	47d 2Fh	2	Hit	{47:506}{47:391}{25:172}{26:172}	

$$\text{Hit-rate:} = \frac{6}{11}$$

- c) El *Hit-rate* de la caché suponiendo que es del tipo *Asociativa por Conjuntos de 2 Vías* es menor a 6/11. Veamos entonces si el *Hit-rate* de la caché asumiendo que es de *Correspondencia Directa* es cómo se indica en el sitio web del fabricante:

direccion	tag (8 bits)	linea (7 bits)	index (5 bits)	resultado	estado cache	notas
CD5E4	205	47	4	Miss	{47:205}	cargué 47:205
CD5E6	205	47	6	Hit	{47:205}	
5633F	86	25	31	Miss	{25:86}{26:86}{47:205}	acceso desalineado cargué 25:86 y 26:86
C3DE9	195	111	9	Miss	{111:195}{25:86}{26:86}{47:205}	cargué 111:195
C3DE1	195	111	1	Hit	{111:195}{25:86}{26:86}{47:205}	
CD5E2	205	47	2	Hit	{47:205}{111:195}{25:86}{26:86}	
5633F	86	25	31	Hit	{25:86}{47:205}{111:195}{26:86}	acceso desalineado
56336	86	25	22	Hit	{25:86}{47:205}{111:195}{26:86}	
FD5EB	253	47	11	Miss	{47:253}{25:86}{111:195}{26:86}	desalojé 47:205 y cargué 47:253
C3DEF	195	111	15	Hit	{111:195}{47:253}{25:86}{26:86}	
FD5E2	253	47	2	Hit	{47:253}{111:195}{25:86}{26:86}	

$$\text{Hit-rate:} = \frac{7}{11}$$

El *Hit-rate* de la caché, asumiendo que es del tipo de *Correspondencia Directa*, es 7/11. Luego, basándonos en la información del sitio web del fabricante podemos concluir que la caché de la máquina en cuestión es de *Correspondencia Directa*.

Ejercicio 4 *ejercicio3.tex* (25 puntos) Debido a la gran exposición a los tsunamis y su relativamente alta probabilidad de ocurrencia es que en Sumatra decidieron instalar un sistema de detección y alerta temprana de dichos eventos naturales. Para ello, se construirá un Sistema de Alerta Local (SAL) utilizando una máquina ORGA1i conectada a un sensor de vibraciones del fondo marino (FONDO) colocado a 200 km de la costa, un sensor de variación de la marea (SUPERFICIE) ubicado en la costa, una **alarma** costera (ALARMA) y un dispositivo encargado de enviar una señal de **alerta** a todas las emisoras televisivas y radiales para que avisen inmediatamente a la población (EMISOR).

La ALARMA comienza a sonar cuando se escribe un 1 en su registro CONTROL_A. Similarmente, el EMISOR emitirá el aviso de alerta cuando se escriba un 1 en su registro CONTROL_D. Por su parte, la magnitud de las vibraciones del fondo marino pueden leerse del registro VIB_FND del sensor de vibraciones FONDO. El nivel de las variaciones puede leerse del registro VAR_SUP del dispositivo SUPERFICIE.

En la siguiente tabla se detalla cómo deben activarse los avisos. PH es el valor del promedio histórico de variación de superficie, ese dato se encuentra en la posición de memoria 0x2000.

	VIB_FND < 0100h	0100h ≤ VIB_FND < 0200h	0200h ≤ VIB_FND
VAR_SUP < PH + 5	ninguno	difundir alerta	alerta y alarma
PH + 5 ≤ VAR_SUP < PH + 10	difundir alerta	activar alarma	alerta y alarma
PH + 10 ≤ VAR_SUP	difundir alerta y activar alarma		

Además, si el centro de estudios sísmicos quiere saber el dato de vibración y marea puede enviar un mensaje por radio al SAL. El radioreceptor (RADIO) del SAL emite una interrupción al recibir el mensaje, que sólo debe ser atendida si no se ha detectado la necesidad de encender la alarma, ni de difundir la alerta. En otro caso, debe ser ignorada. Los valores de ambos sensores deben ser guardados en R1 y R2 para que luego estos sean leídos por la rutina ENVIARDATOS, ubicada en el rango de memoria 0x1000 a 1FFF.

- Dibujar un diagrama que muestre la conexión de **todos** los componentes del SAL (cpu, memoria y dispositivos). Para cada registro de E/S, indicar su ubicación (en el diagrama), decidir si es de escritura, lectura o ambas y realizar los *mapeos* de direcciones necesarios.
- Explicar² la rutina principal del SAL que se encarga de leer continuamente los sensores y actuar de acuerdo a sus resultados. Luego, escribirla en lenguaje ensamblador.
- Explicar² la rutina de atención de interrupciones. Puede asumir que se cuenta con una etiqueta `enviarDatos` que indica el comienzo de la rutina que envía la información en R1 y R2 a la central de operaciones. Luego, escribirla en lenguaje ensamblador.

Solución.

- (Falta el diagrama.) *Mapeos* de las direcciones de E/S en direcciones de memoria:

- CONTROL_A (Alarma): 0xFFFF0 - Escritura
- CONTROL_D (Emisor): 0xFFFF1 - Escritura
- VIB_FND (Fondo): 0xFFFF2 - Lectura
- VAR_SUP (Superficie): 0xFFFF3 - Lectura

- MAIN:

```

CLI ; Deshabilito las interrupciones hasta tanto no chequee que no haya que prender
    ; la alarma o la alerta
MOV R1, [0x2000]
ADD R1, 5 ; En R1 guardo PH + 5
MOV R2, R1
ADD R2, 5 ; En R2 guardo PH + 10
LOOP:
CLI ; Deshabilito las interrupciones porque el ciclo vuelve a LOOP, no a MAIN
CMP [0xFFFF2], 0x0200
JGE ACTIVAR_ALARMA ; VIB_FND >= 0x0200 ---> Activo ambas alarmas

```

²NOTA: Para explicar las rutinas puede utilizar *pseudocódigo*.

```

    CMP [0xFFF3], R2
    JGE ACTIVAR_ALARMA ; VAR_SUP >= PH + 10 ---> Activo ambas alarmas
    CMP [0xFFF2], 0x0100
    JGE CHEQUEAR_ACTIVAR_ALARMA
    CMP [0xFFF3], R1
    JGE ACTIVAR_ALERTA ; PH + 10 > VAR_SUP >= PH + 5 && 0x0100 > VIB_FND ---> Activo alerta
    STI ; Recién ahora me pueden interrumpir
    JMP FIN
CHEQUEAR_ACTIVAR_ALARMA:
    CMP [0xFFF3], R1
    JL ACTIVAR_ALERTA ; PH + 5 > VAR_SUP && 0x0200 > VIB_FND >= 0x0100 ---> Activo alerta
    MOV [0xFFF0], 0x0001 ; PH + 10 > VAR_SUP >= PH + 5 && 0x0200 > VIB_FND >= 0x0100
    ; ---> Activo alarma
    STI ; Recién ahora me pueden interrumpir (según el enunciado este podía no ponerlo)
    JMP FIN
ACTIVAR_ALERTA:
    MOV [0xFFF1], 0x0001
    STI ; Recién ahora me pueden interrumpir (según el enunciado este podía no ponerlo)
    JMP FIN
ACTIVAR_AMBOS:
    MOV [0xFFF0], 0x0001
    MOV [0xFFF1], 0x0001
    STI ; Recién ahora me pueden interrumpir (según el enunciado este podía no ponerlo)
FIN:
    JMP LOOP

```

c) RAI:

```

    PUSH R1
    PUSH R2
    MOV R1, 0xFFF2
    MOV R2, 0xFFF3
    CLI
    CALL enviarDatos
    STI
    POP R2
    POP R1
    IRET

```
