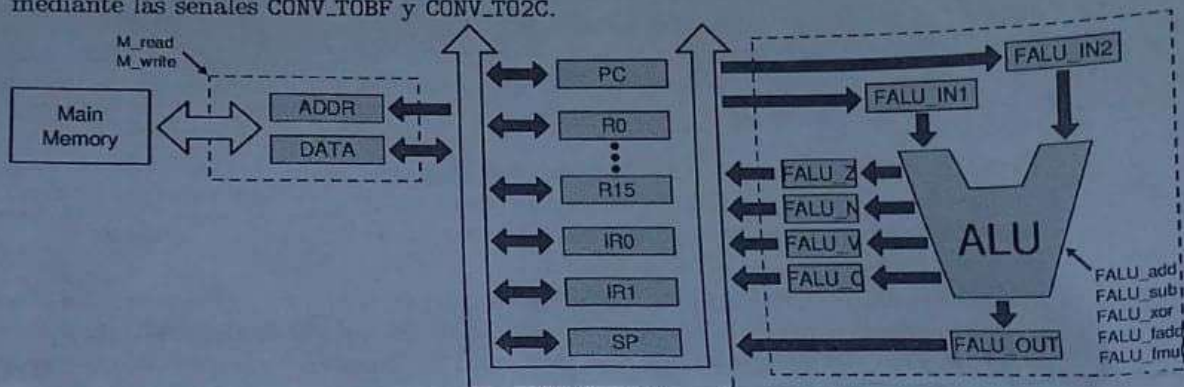


Ejercicio 1 Microprogramación

Se desea agregar soporte de punto flotante a una microarquitectura con palabras, registros y direcciones de 16 bits e instrucciones de una o dos palabras. Para ello, se extiende la ALU con operaciones de punto flotante y los registros R8 a R15 se destinan a almacenar valores *bfloat16*¹. También se dispone de un componente CONVERTER con registros CONV_IN de entrada y CONV_OUT de salida que permite convertir valores en complemento a dos a valores *bfloat16* y vice-versa mediante las señales CONV_TOBF y CONV_TO2C.



El procesador resuelve las instrucciones:

- De dos parámetros: ADD, SUB, XOR, MOV, FADD, FMUL. Todas salvo MOV modifican los flags.
- De un parámetro: JMP, JZ, JC, JV, JN (desplazamiento inmediato)

Para los modos de direccionamiento:

- Directo a registro: Rxx
- Inmediato: imm16 (en complemento a dos)
- Indirecto a registro: [Rxx]

Cualquier combinación de modos es posible salvo entre dos inmediatos.

Para las operaciones aritméticas es obligatorio respetar el tipo de los operandos: ADD/SUB toman y devuelven enteros en complemento a dos, mientras que FADD/FMUL toman y devuelven *bfloat16*. Si algún operando proviene de un registro de distinto tipo al esperado, deberá ser convertido al formato correcto; lo mismo para el registro destino de la operación. Esto vale al mover datos entre registros de distinto tipo. Si un operando proviene de la memoria, se asume que está en el formato esperado.

1. Proponer una codificación variable de una o dos palabras para este set de instrucciones.
2. Completar el *datapath* e indicar el tamaño y conectividad de cada registro.
3. Explicar cómo se implementa el fetch de instrucciones sobre el *datapath* propuesto.
4. Codificar las siguientes instrucciones e indicar su secuencia de microinstrucciones:
 - ADD R8, [R1]
 - FMUL R8, -4
 - JZ 0x016
 - FADD R8, R9
 - FADD R1, R2
 - JN 0xFF0

Ejercicio 2 Memoria Cache

Sea la siguiente lista de accesos a memoria en una máquina con 1 KB de memoria direccionable a byte: 0x002, 0x082, 0x102, 0x182, 0x004, 0x084, 0x104 y 0x184. En cada caso, el acceso realizado es una lectura de 4 bytes.

¹Equivalente a IEEE-754 de 32 bits truncado a 16 (un bit de signo, 8 de exponente y 7 de mantisa).

1. Calcular el *hit-rate* para memorias caché de los tipos indicados a continuación:

- Mapeo directo.
- Asociativa por conjuntos de 2 vías (LRU).

En ambos casos, las cachés tienen 32 bytes de tamaño (sin contar el almacenamiento para los tags) y líneas de 4 bytes. Indicar para cada acceso si se realizó un *miss* o un *hit* y cuál es el estado completo de la caché. Además detallar si se realizó algún acceso desalineado.

2. Suponiendo que no es posible modificar el tamaño total de la caché, ¿existe alguna configuración (tipo, política de reemplazo si corresponde y tamaño de línea) que mejore los *hit-rates* obtenidos en el ítem anterior? ¿Cuál es el máximo *rate* alcanzable? Justificar.

Ejercicio 3 Entrada/Salida

Las baterías de notebook de litio-ión sufren de un proceso acelerado de desgaste cuando son sometidas a ciclos de descarga completos (cuando son llevadas del 100% al 0% y luego vueltas a cargar). Por este motivo, las notebooks incorporaron un controlador en la batería para regular el nivel de carga cuando la máquina se encuentra conectada a la corriente eléctrica.

Se tiene una notebook con un procesador 8086 (de última generación) y un PIC 8259 en el cual están conectados una batería (IRQ 5, direcciones de E/S BATERIA y NIVEL) y un dispositivo de apagado forzado (IRQ 4, dirección de E/S APAGADO). Este último está conectado directamente a la batería por medio de una línea dedicada. Los registros de E/S son de lectoescritura de 16 bits, y codifican la siguiente información:

15 14	8 7	0
C	Reservado	Carga

BATERIA

15	8 7	0
Máximo	Mínimo	

NIVEL

15 14	8 7	0
A	Reservado	Umbral

APAGADO

- BATERIA: el bit C indica 1 si debe cargarse, 0 si debe descargarse². Los bits Carga indican el nivel de carga de la batería como número entre 0 y 100. Los bits Reservado no deben modificarse.
- NIVEL: se codifican en Máximo y Mínimo números entre 0 y 100 que indican entre qué niveles se debe mantener la carga de la batería. Si alguno de dichos valores es alcanzado (el máximo en la fase de carga y el mínimo en la fase de descarga), el dispositivo generará una interrupción.
- APAGADO: el bit A indica que debe apagar el procesador (cuando se le escribe el valor 1). Los bits Umbral codifican con un número entre 0 y 100 el umbral crítico de carga por debajo del cual este dispositivo generará una interrupción. Los bits Reservado no deben modificarse.

Se pide:

1. Escribir en assembler de 8086 la inicialización de ambos dispositivos, configurando la batería para que mantenga su carga entre 30 y 80% y que la notebook se apague al caer por debajo del 10%. Puede suponer que la batería tiene 100% de carga y está conectada a la corriente.
2. Utilizando *polling*, y suponiendo que las interrupciones se encuentran desactivadas, escribir una rutina en assembler de 8086 que intente mantener la carga de la notebook dentro de los valores configurados, y que apague la máquina en caso de que la misma caiga por debajo del umbral crítico. Además, almacenar en memoria principal la cantidad de veces que se completa un ciclo de descarga parcial (veces que la carga pasó del nivel máximo indicado en NIVEL al mínimo) y de carga parcial (del mínimo al máximo).
3. Modificar el código anterior para que utilice la funcionalidad provista por el PIC. Indicar cómo se debe configurar este último para interactuar correctamente con el 8086. El dispositivo de apagado debe poder interrumpir con **mayor prioridad** que la batería.

Ejercicio 4 Buses

Sea una arquitectura con registros de 32 bits y direcciones de 24 bits. Diseñar un bus sincrónico para interconectar el procesador con un módulo de memoria y un dispositivo de E/S mapeado a memoria. Se cuenta con 16 líneas que será necesario distribuir entre señales de todo tipo.

1. Definir cada una de las señales necesarias, su tipo, y si son o no multiplexadas.
2. Dar el diagrama de tiempos de una operación de escritura a memoria, suponiendo que ésta necesita un ciclo de reloj para preparar la escritura una vez recibida la dirección.
3. Dar el diagrama de tiempos de una operación de lectura a dispositivo, asumiendo que éste puede tardar una cantidad arbitraria de ciclos en responder con el dato. Explicar detalladamente la secuencia de estados atravesada por la CPU y el dispositivo.

²Notar que el valor de dicho bit no necesariamente coincide con el estado de carga real de la batería, ya que la misma podría no estar conectada a la corriente.