

Representación de la información

Estamos desarrollando un algoritmo detector de patrones audio. Para ello, capturamos la señal de una placa de audio profesional a 192 kHz (192000 muestras/seg), con una resolución de 16 bits por muestra. Dado que para que el algoritmo funcione adecuadamente necesitamos una señal de bajo ruido, pero son suficiente 12000 muestras/seg, decidimos hacer un promediador de 16 muestras (esto es, por cada 16 muestras las promediamos y nos quedamos con ese valor).

Si las muestras las tenemos representadas en `int16_t`:

- a. Si por razones de implementación sólo podemos utilizar enteros de 16 bits: ¿Cómo implementaríamos ese promediador si necesitamos evitar errores por saturación? ¿Qué problemas numéricos se presentan?
- b. Cuándo no hay señal a la entrada, tenemos la siguiente secuencia de muestras:

0x0023, 0xFFFE, 0xFFFF1, 0x001B, 0x003A, 0xFFE0, 0xFFDA, 0x000B,
0x003A, 0xFFFF7, 0xFFFF0, 0x0013, 0x000F, 0xFFE5, 0xFFE6, 0x0033

¿Cuánto dará el promedio propuesto en el punto a?

- c. Si transformáramos las muestras a punto flotante de precisión simple (float), y sabiendo que las muestras cambian muy lento (es decir las muestras vecinas tienen siempre valores similares en el mismo rango) ¿convendría usar el mismo algoritmo que en el punto a. para promediar? ¿Por qué?

Microarquitectura

Se tiene una arquitectura con el siguiente esquema:

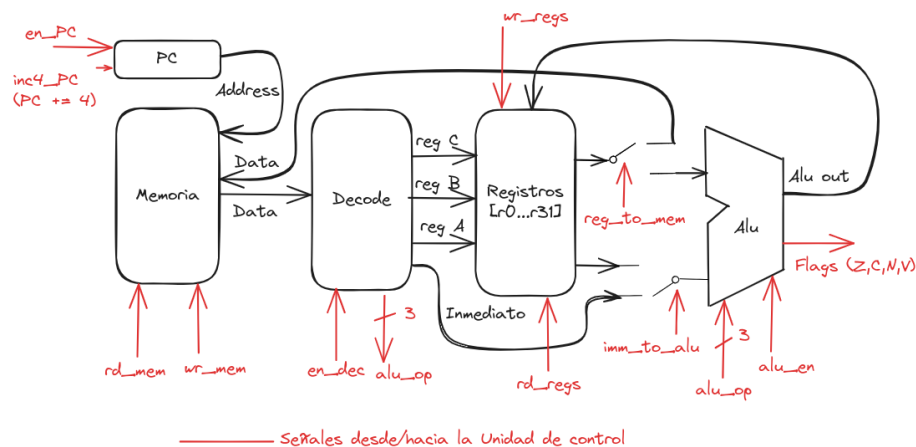


Figure 1: Esquema de la arquitectura

Con las siguientes características: - Tamaño de palabra: 32 bits - Tamaño de direcciones: 32 bits - Tamaño de todas las instrucciones: 32 bits

Se asume que todas los módulos internos tardan un clock en actualizar sus salidas.

Se pide:

1. Dar la secuencia de micro-instrucciones necesarias para ejecutar la siguiente instrucción:

add r3, r2, r1

Dónde **r3** es el registro destino, **r2** es el registro fuente 2 y **r1** es el registro fuente 1.

- Indicar claramente a qué etapa del ciclo de instrucción se corresponde cada grupo de micro-instrucciones (Se asume que el PC se encuentra apuntando internamente a la dirección dónde se encuentra la instrucción a implementar).
2. ¿Es posible implementar instrucciones de salto (condicional/incondicional) en esta arquitectura? ¿Por qué? Justificar indicando la forma de hacerlos o introduciendo las modificaciones necesarias al esquema.

Memoria Cache

Hacer el seguimiento de escrituras y lecturas de la siguiente secuencia de direcciones, en una memoria cache con las siguientes características: Dirección de 5 bits y memoria direccionable a byte.

Características de la memoria:

- 4 sets
- 2 líneas por set
- 2 bytes por bloque
- 1 byte por lectura/escritura

Suponga que la memoria cache está vacía al inicio y utiliza como política write-back / write-allocate. La política de desalojo es LRU.

op	address	t	s	b	hit/miss/dirty-miss
W	0x10				
W	0x08				
R	0x18				
R	0x02				
R	0x11				
W	0x1C				
R	0x19				
W	0x19				
R	0x03				

Assembly

Implementar la siguiente función en assembly:

Dada una lista enlazada, insertarle un nuevo nodo al final de la lista que tenga como dato la suma de todos los demás nodos de la lista. Se puede considerar que la lista tendrá por lo menos un elemento.

```
void sumaElemento(list_t* lista)
```

Donde la estructura de lista y de su nodo es:

```
typedef struct s_list {  
    int size;  
    struct s_listElem* first;  
    struct s_listElem* last;  
} list_t;
```

```
typedef struct s_listElem {  
    struct s_listElem* next;  
    int data;  
} listElem_t;
```