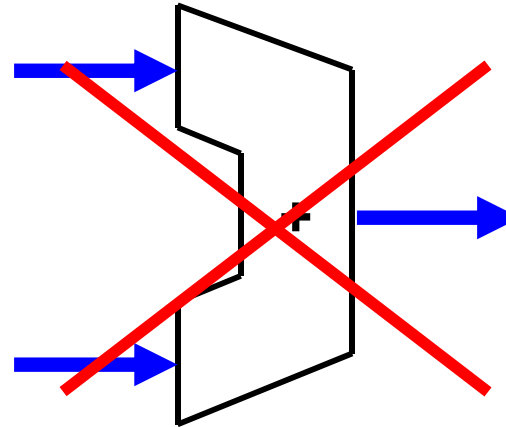


Microprocesador simplificado (sumador)

Circuito sumador genérico

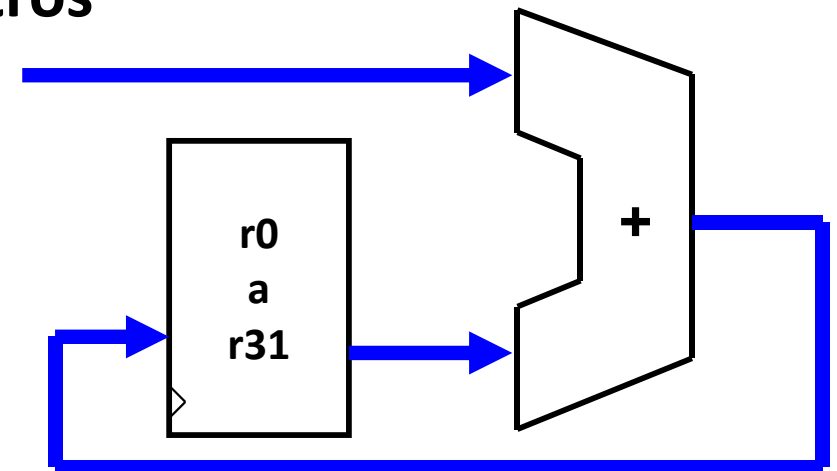
Nos planteamos realizar un circuito para sumar cualquier número de operandos (un sumador de un número fijo de operandos no sirve):

- ✓ $A + B$;
- ✓ $C + D + E$;
- ✓ $F + G + H + I$;
- ✓ ...



Sirve un sumador de dos entradas si almacenamos resultados parciales en registros => **banco de registros**

- ✓ $R_1 = R_0 + F$;
 - ✓ $R_2 = R_1 + G$;
 - ✓ $R_3 = R_2 + H$;
 - ✓ $R_4 = R_3 + I$;
- Equivalente a:
 $R_4 = F + G + H + I$;
(el reg. R_0 es siempre 0)



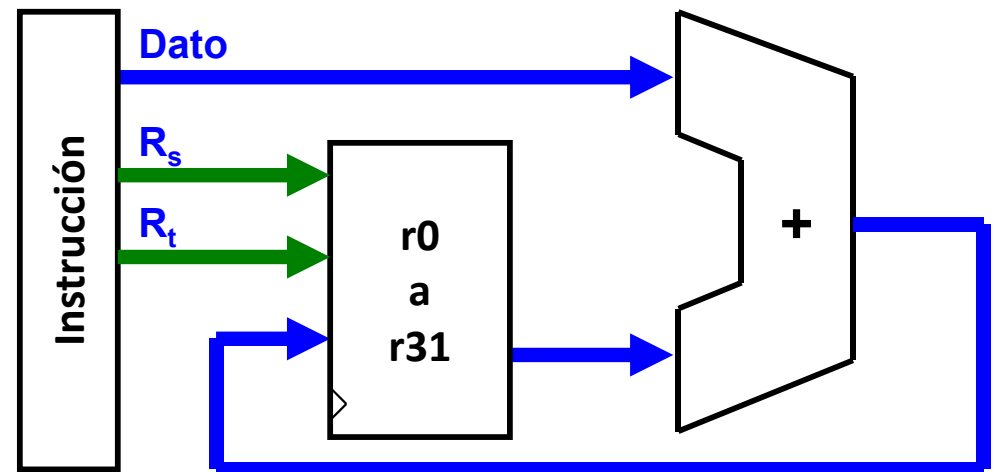
Circuito sumador genérico

Por tanto, necesitamos un circuito capaz de realizar la siguiente “instrucción” (es un microprocesador muy simplificado):

$$✓ R_t \leftarrow R_s + \text{Dato};$$

En cada instrucción necesitamos darle la siguiente información:

- ✓ Número del registro destino (R_t), del 0 al 31 => 5 bits
- ✓ Número del registro fuente (R_s), del 0 al 31 => 5 bits
- ✓ Dato (dato inmediato) => 16 bits



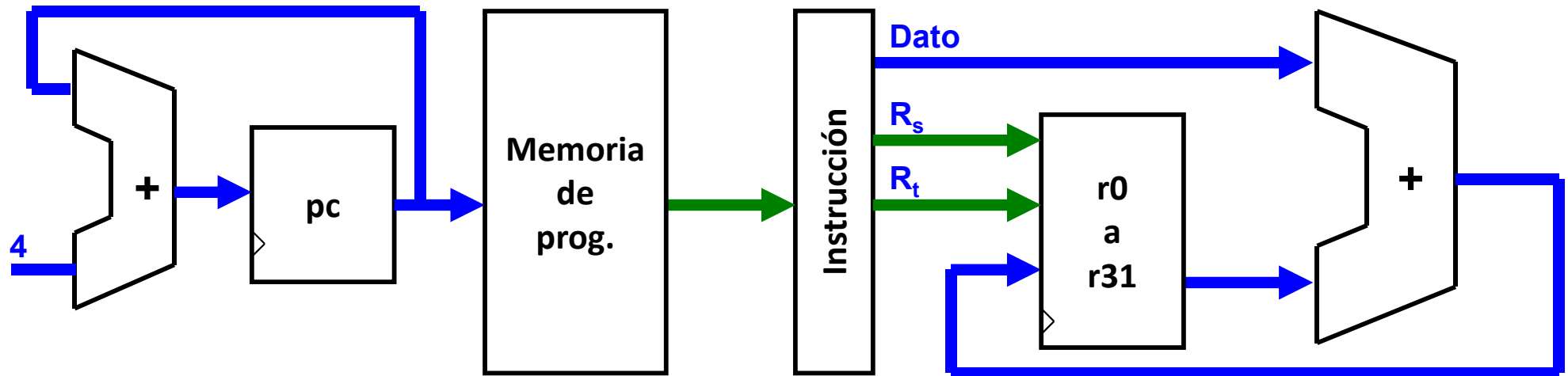
Circuito sumador genérico

¿Cómo recibe el micro las instrucciones?

- ✓ Se almacenan en una memoria (programa)
- ✓ El micro tiene que ser capaz de ir leyendo la memoria, instrucción tras instrucción

Cada instrucción necesita $5+5+16 = 26$ bits

- ✓ Se ajusta a 32 (potencia de 2)
- ✓ 32 bits \Rightarrow 4 bytes, cada instrucción está 4 bytes más adelante



Ejemplo de funcionamiento

Sumar $8 + 21 + 14$. Se hace con tres instrucciones:

✓ $R_1 = R_0 + 8;$

✓ $R_2 = R_1 + 21;$

✓ $R_3 = R_2 + 14;$

El contador de programa (*program counter*, pc) indica la dirección de memoria de la instrucción actual.

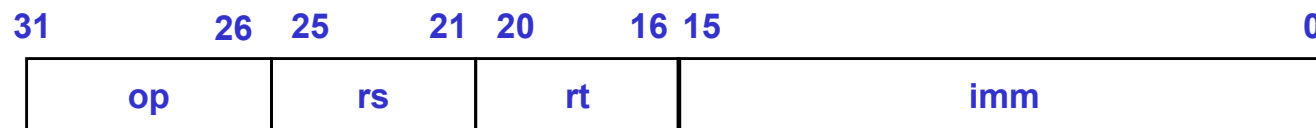
✓ Empieza en 0 y sube 4 cada instrucción

Dirección		Memoria de programa
0		$R_1 = R_0 + 8;$
4		$R_2 = R_1 + 21;$
8		$R_3 = R_2 + 14;$
C		????
...		
FF...C		????

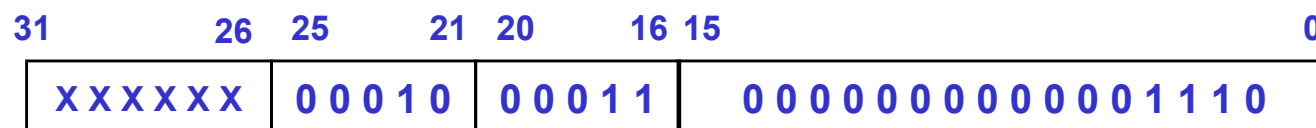
Memoria de programa (instrucciones)

En cada posición de memoria se almacena una instrucción de 32 bits:

- ✓ 5 bits para el registro destino, R_t
- ✓ 5 bits para el registro fuente, R_s
- ✓ 16 bits para el dato inmediato
- ✓ Resto de bits no se usan (en micros reales indican el tipo o código de instrucción, *operation code* (op), ya que hay más de una)



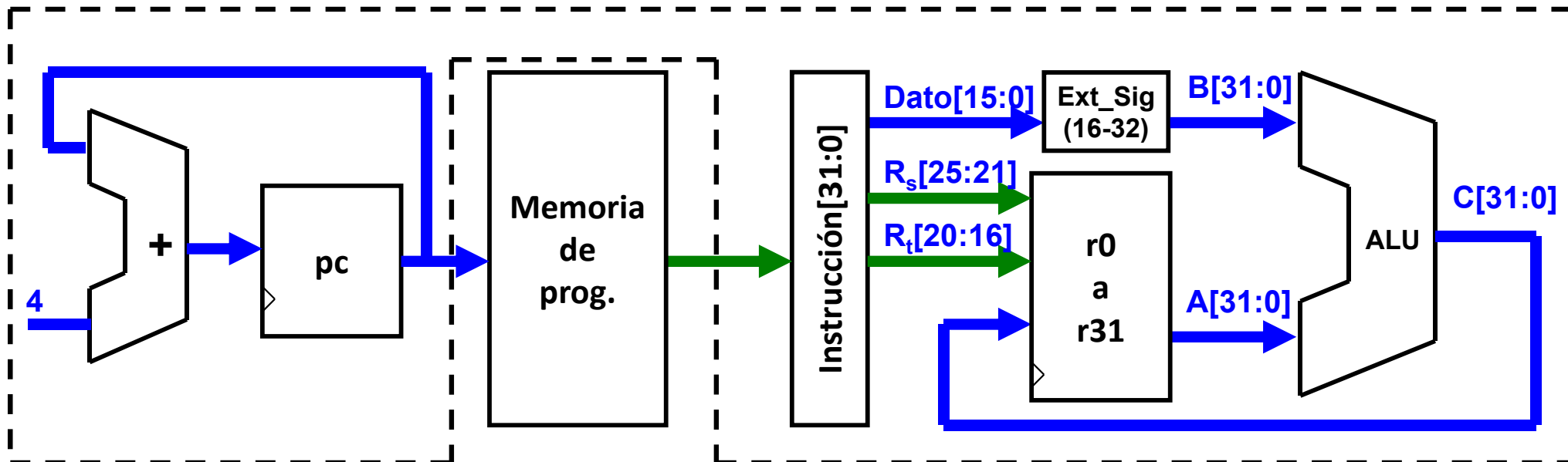
Ejemplo: $R3 = R2 + 14;$



Ancho de palabra

Nuestro microprocesador utilizará datos de 32 bits:

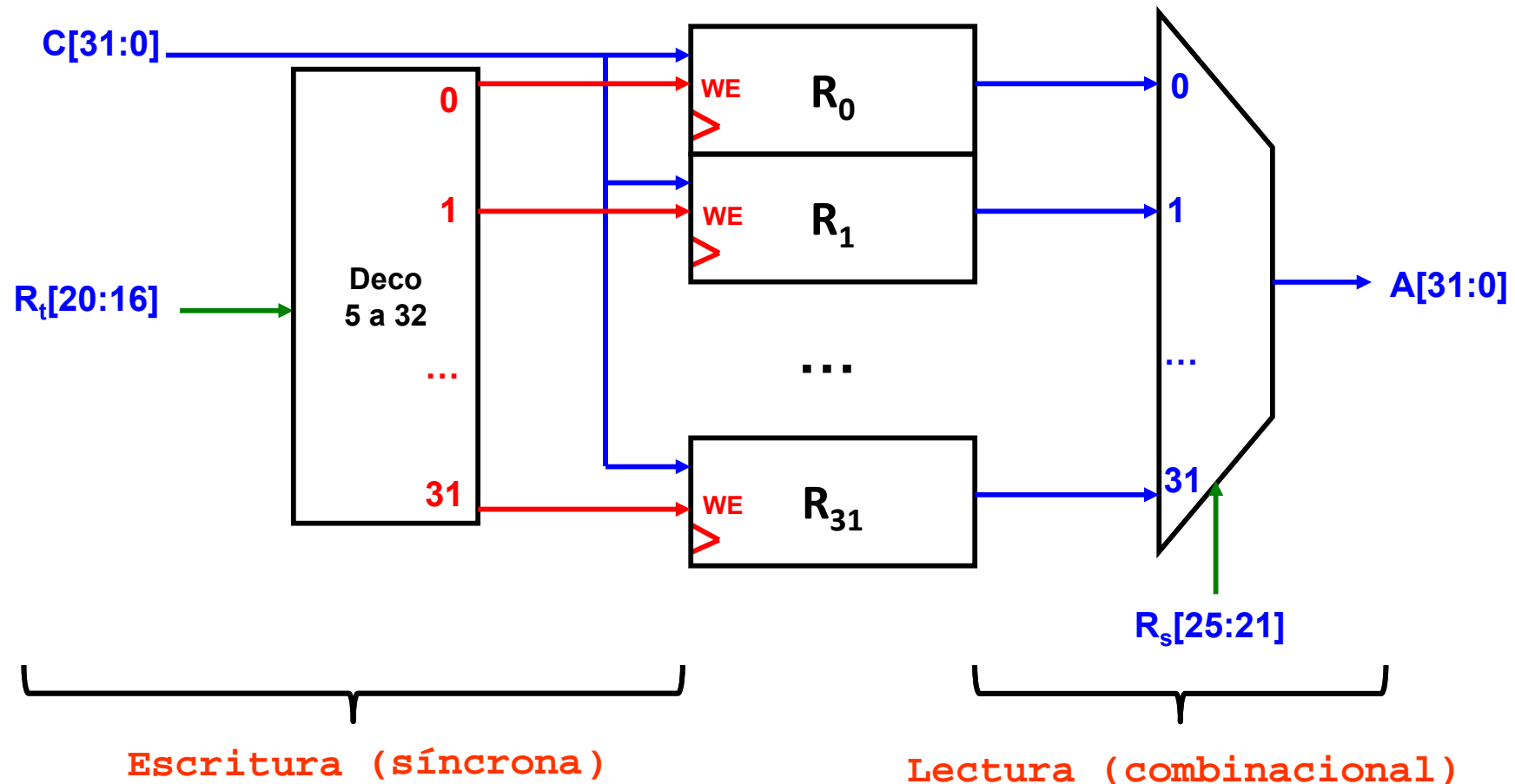
- ✓ Las entradas y salida del sumador (ALU) son de 32 bits
- ✓ Los registros del banco de registros son de 32 bits
- ✓ Como el dato inmediato es de 16 bits, se extiende (con signo) a 32 bits



Banco de registros

¿Cómo se realiza el banco de registros?

✓ Básicamente, multiplexando 32 registros (cada uno de 32 bits)

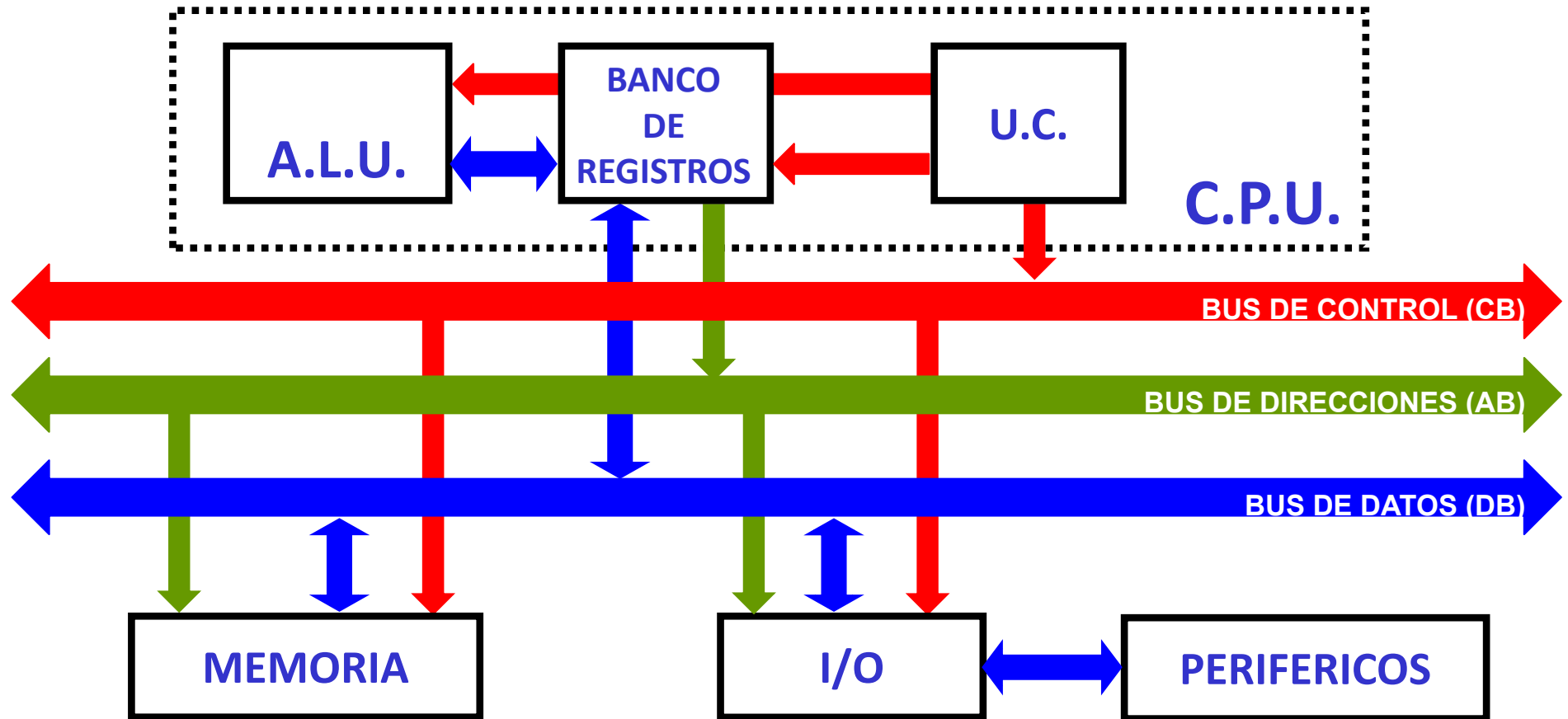


Microprocesador completo

¿Qué le falta para ser un microprocesador completo?

- ✓ Poder realizar otras operaciones (instrucciones aritmético-lógicas)
- ✓ Poder usar más de 32 datos, y para ello se añade la memoria de datos (instrucciones con acceso a memoria de datos)
- ✓ Poder variar la secuencia de ejecución para realizar bucles o control de flujo, como for, if, etc... (saltos)

Arquitectura clásica Von Neumann



Arquitectura Harvard

