

# UNIDAD II

## ABACUS - SUPERABACUS

## ARQUITECTURA VON NEUMANN

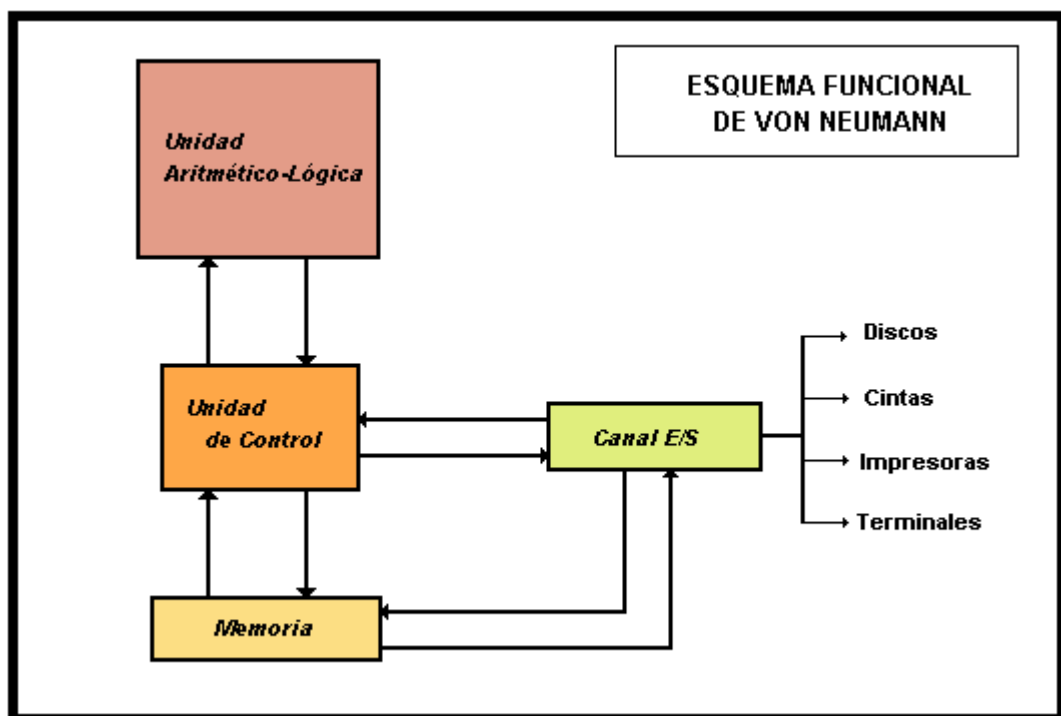
Hasta mediados de los años cuarenta los computadores se programaban por medio de la manipulación de un panel frontal, en el cual los usuarios conectaban o desconectaban determinados switches (conmutadores), para que el computador realizase una u otra función y así llevar a cabo la tarea deseada. Es Von Neumann quien hacia 1945, promueve el paso decisivo hacia la mecanización del tratamiento digital de la información, con la invención de dos nuevos conceptos:

- ***El programa almacenado:*** el computador tendría un programa de instrucciones almacenado en su propia memoria. En lugar de ejecutar las operaciones al compás de su lectura (en una cinta perforada por ejemplo) la nueva maquina supone almacenado en memoria el programa previamente a la ejecución de las operaciones. El uso de memoria es tanto para el almacenamiento de datos como de instrucciones de programa. Esta idea sencilla es la que ha imperado en el diseño de los computadores hasta nuestros días.
- ***La ruptura de Secuencia:*** esta es la clave del poder decisorio de los programas. En lugar de necesitar intervención humana cada vez que se planteaba una toma de decisión (íntimamente ligada a los resultados que se iban obteniendo) las operaciones de decisión lógica serían automáticas dotando a la máquina de una instrucción llamada *salto condicional* o *ruptura condicional de secuencia*. Según el valor de un resultado ya obtenido, la maquina ejecutaría una u otra parte del programa.

Los computadores con una arquitectura Von Neumann pueden ser considerados como un conjunto de unidades conectadas entre sí, con una función determinada dentro del esquema del computador. Constan de cinco partes componentes:

- ***Memoria:*** se encuentra dividida en celdas o posiciones de memoria cuyo contenido es variable y son identificadas por un número fijo llamado dirección de memoria. La capacidad total de una memoria está dada por la cantidad de celdas disponibles, en ella se almacenan dos clases de información: las instrucciones del programa que se deberá ejecutar y los datos (comúnmente llamados operandos) con los cuales deberá trabajar el programa.
- ***Unidad aritmético-lógica (UAL):*** es la unidad encargada de realizar las operaciones elementales de tipo aritmético (sumas y restas) y de tipo lógico (generalmente comparaciones).

- **Unidad de control (UC):** también llamada unidad de gobierno, desde ella se controlan y gobiernan todas las operaciones (búsqueda, decodificación y ejecución de instrucciones). Es la encargada de extraer de la memoria central la nueva instrucción a ejecutar, analizarla y extraer también los operandos implicados. A su vez, desencadena el tratamiento de los datos en la UAL y, de ser necesario, los almacena en la memoria central.
- **Dispositivo de entrada/salida:** gestiona la transferencia de un conjunto de informaciones entre las unidades periféricas y la memoria central tanto en un sentido como en otro, y es su tarea advertir a la unidad de control el momento en el cual todas las informaciones han sido transferidas.
- **Bus de datos:** es un sistema digital que proporciona un medio de transporte de datos entre las distintas partes, no almacena información sólo la transmite.



En resumen se puede representar un ordenador como un conjunto ensamblado de unidades diferentes, cuyo funcionamiento viene dictado por el programa registrado en la memoria central. La unidad de control gobierna la ejecución de las operaciones pedidas por dicho programa. Si la operación es un cálculo, es la unidad aritmético-lógica quien lo realiza, si es una transferencia de informaciones con el exterior se cede el control a un dispositivo de E/S o canal.

## **CONCEPTOS PRELIMINARES**

Previo análisis de la máquina Abacus será necesario introducir las siguientes definiciones que se encuentran íntimamente ligadas entre sí y son de vital importancia para comprender el funcionamiento de un computador:

**Registro:** los registros son considerados los bloques más importantes de un computador. Una definición general los identifica como una “memoria muy rápida” que permite almacenar una cierta cantidad de bits (información).

**Compuerta:** las compuertas constituyen la base del hardware sobre la cual se construyen los computadores digitales. Son circuitos electrónicos biestables unidireccionales, es decir, permiten el pasaje de información en un sólo sentido y admiten únicamente dos estados (abierto / cerrado, 1 / 0).

**Bus:** si bien se aproximó una definición al comienzo de este apunte, podemos profundizar aun más e identificar distintos tipos de buses, a saber:

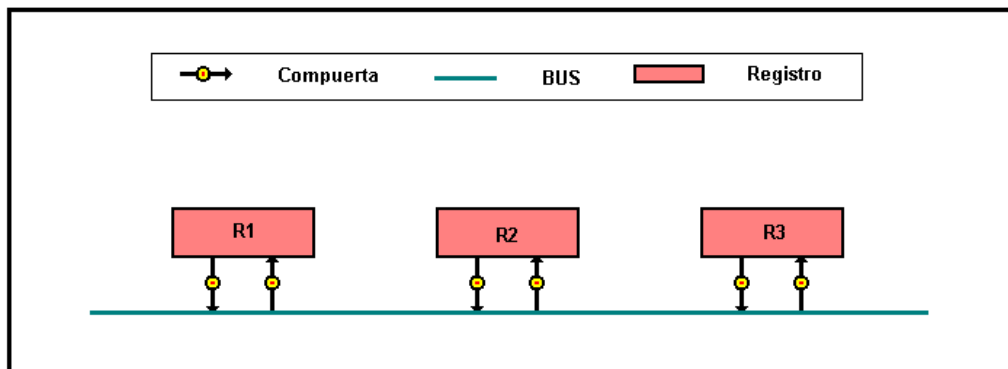
Bus de Datos: mueve la información por los componentes de hardware internos y externos del sistema tanto de entrada como de salida (teclado, Mouse etc.)

Bus de Direcciones: ubica los datos en la memoria teniendo relación directa con los procesos de la CPU.

Bus de Control: marca el estado de una instrucción que fue dada a la PC.

La transferencia de datos puede darse entre los componentes de un ordenador o entre ordenadores, es decir existen buses internos (ej: transportar datos desde y hasta la UAL) y buses externos (ej: conectar el ordenador con dispositivos de E/S).

Los tres elementos que hemos definido (Compuerta, Registro y Bus) nos permitirán comprender cómo se lleva a cabo la transferencia de información y, por ende, el proceso de ejecución de una instrucción.



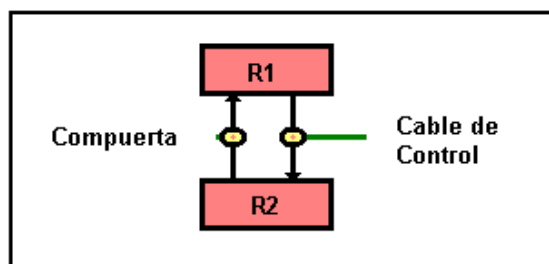
Para intercambiar información entre dos registros, es necesario que dicha información viaje a través del bus. El traspaso de la misma entre un registro y el bus de datos, por ejemplo,

### 75.03 Organización del Computador

esta regido por las compuertas: sólo si la compuerta está abierta se produce el pasaje de datos. Cuando se abre una compuerta la información esta disponible en el bus y mientras esté abierta los datos permanecerán en el mismo. Sin embargo el bus no tiene capacidad de almacenamiento, por ende al cerrar la compuerta la información desaparecerá.

La compuerta que manda información al bus es única, es decir, no pueden existir dos abiertas simultáneamente, esto es importante tenerlo presente a la hora de indicar la secuencia de apertura de compuertas en el proceso de ejecución de una instrucción.

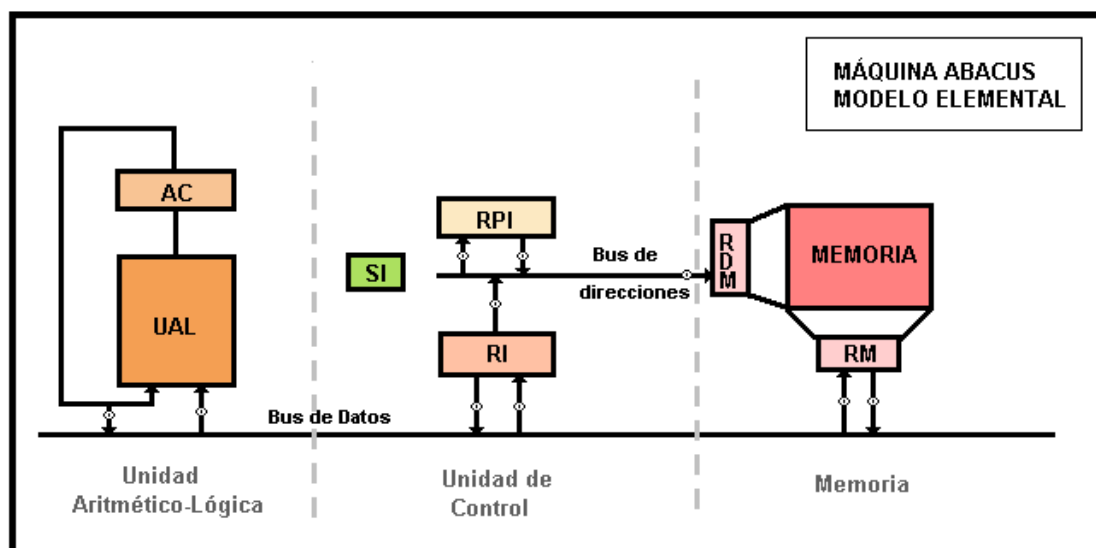
Tomemos la siguiente imagen como ejemplo del traspaso de información entre registros:



Al abrir la compuerta se copia la información de R2 a R1, al cerrarla la información queda almacenada en ambos registros. El cable de control es quien da la orden de abrir o cerrar la compuerta (que permitirá el pasaje unidireccional de datos).

### ABACUS: MAQUINA ELEMENTAL

Abacus es una maquina de tipo Von Neumann cuyo esquema es el siguiente:



#### Registros

**AC:** acumulador

**SI:** secuenciador de instrucciones

**RI:** registro de instrucción

**RM:** registro de memoria

**RI:** registro de instrucción

**RPI:** registro de próxima instrucción

**RDM:** registro de direcciones de memoria

## Descripción de Componentes

### I) UAL:

Abacus es una máquina de una sola dirección, su unidad aritmético-lógica posee un registro particular llamado acumulador (**AC**) que sirve tanto para albergar el primer operando como para albergar el resultado. Esta característica permite instrucciones de una sola dirección: la del segundo operando.

Entre las operaciones que realiza la UAL pueden mencionarse:

- CARGA (cargar un contenido en el AC)
- ALMACENAMIENTO (enviar datos a memoria por el bus)
- SUMA (sumar un dato a lo que haya en el AC)
- LOGICAS (OR, AND ,XOR)

Es importante recordar que en una maquina Abacus **todas las operaciones** se realizan “contra” el acumulador y el resultado siempre queda almacenado en él.

### II) UC

La Unidad de Control, como ya hemos visto, es la encargada de extraer y analizar las instrucciones de la memoria central y, para ello, se vale de dos registros:

**RPI:** contiene la dirección de la próxima instrucción a ejecutar, se comunica con la memoria y con el RI a través del Bus de Direcciones. A medida que se van ejecutando las instrucciones este registro aumenta su contenido en una unidad excepto para las instrucciones de ruptura de secuencia.

**RI:** contiene la instrucción extraída de la memoria, en ella podemos identificar dos partes fundamentales: el código de operación y la dirección del operando. Este registro se comunica con la memoria mediante el Bus de Datos.

En cuanto al **SI** cabe destacar que tras analizar el código de operación distribuye las ordenes de la Unidad de Control a la Unidad Aritmético-Lógica y a la Memoria para ejecutar las fases de la instrucción, en otras palabras, es el encargado de administrar la apertura y cierre de las compuertas para el correcto funcionamiento y pasaje de la información.

### III) MEMORIA

El intercambio de información entre la memoria y el resto del computador, ya sea para acceder a los datos o para almacenar información en la misma, se lleva a cabo a través de dos registros:

**RDM:** contiene la dirección de la celda de memoria en (de) la cual se escribirá (leerá) la información

**RM:** contiene el dato que debe ser leído (escrito) desde (en) la memoria.

### Registro de Instrucción

Vamos a detenernos un momento en un concepto clave de la Máquina Abacus: el formato de instrucciones que soporta:

<b>CO</b>	<b>Operando</b>
-----------	-----------------

Para entender a que hacemos referencia cuando decimos que Abacus es una máquina de una sola dirección tomemos como ejemplo una instrucción SUMA, el objetivo es sumar el contenido de dos celdas de memoria distintas (de diferente dirección) y almacenar el resultado en una de ellas.

Dado que en la instrucción sólo podemos indicar la dirección de un único operando nos veremos obligados a realizar las siguientes acciones:

1. Cargar el primer operando en el AC.
2. Sumar el segundo operando al contenido del AC.
3. Almacenar en memoria el resultado contenido del AC.

Se deduce a simple vista que todas las operaciones se realizan “contra el acumulador”.

<b>CARGAR</b>	<b>200</b>	<b>Se suman los contenidos de las celdas de memoria cuyas direcciones son 200 y 206</b> <b>El resultado se almacena en la celda de memoria de direccion 200</b>
<b>SUMAR</b>	<b>206</b>	
<b>ALMACENAR</b>	<b>200</b>	

En este ejemplo vemos que de ser posible indicar dos direcciones, bastaría tan solo con ejecutar una única instrucción que contenga las direcciones de los dos operandos que queremos sumar.

### Relaciones

En base a las relaciones que se establecen entre los componentes de la Máquina Abacus se pueden deducir las siguientes equivalencias:

- Tamaño RPI = Tamaño RDM = Tamaño Op = Cantidad de Celdas Direccionables
- Tamaño AC = Tamaño RI = Tamaño RM = Longitud de Instrucción = Longitud de Celda

Esto se verá claramente a continuación, donde detallaremos el desarrollo de una instrucción Abacus.

## Desarrollo de una instrucción

El desarrollo de una instrucción en Abacus puede descomponerse en 2 fases:

- **Fase de Búsqueda:** consiste en localizar la instrucción que se va a ejecutar; esta fase es común a todos los tipos de instrucciones y, a su vez, la secuencia de acciones que se lleva a cabo es idéntica a la búsqueda de operandos. También se actualiza en forma secuencial la dirección de la siguiente instrucción a ejecutar.
- **Fase de Ejecución:** como su nombre lo indica consiste en ejecutar la instrucción, claramente esta fase es dependiente del tipo de tarea a realizar (suma, almacenamiento, salto, etc.).

### Descripción de Fases

**I) Búsqueda de una instrucción:** La unidad de control ordena la transferencia del contenido del RPI al RDM y envía a la memoria una orden de lectura. El contenido de la celda de memoria queda almacenado en el RM, luego la Unidad de Control ordena la transferencia del contenido del RM al RI, pudiendo entonces los circuitos especializados analizar el código de operación de la instrucción. Finalmente se prepara el RPI para ejecutar la próxima instrucción.

$$\begin{array}{ll} \text{RDM} & \longleftarrow (\text{RPI}) \\ \text{RM} & \longleftarrow ((\text{RDM})) \\ \text{RI} & \longleftarrow \text{RM} \\ \text{RPI} & \longleftarrow (\text{RPI}) + 1 \end{array}$$

Para realizar la búsqueda de un operando, una vez que la Unidad de control analiza el contenido del código de operación, ordena la transferencia del contenido del campo operando del RI al RDM y luego envía una orden de operación de lectura. El operando buscado queda disponible en el RM.

**II) Ejecución de una instrucción:** la ejecución de cada instrucción implica el movimiento de los datos y como estos pasos se realizan en forma secuencial y ordenada la UCP sigue las señales dadas por el reloj del sistema. Se ha indicado más arriba que esta fase depende exclusivamente del tipo de tarea que se deba realizar, por ende se analizarán casos particulares.

- **Suma:** se debe sumar el contenido del RM al contenido del acumulador

$$\begin{array}{ll} \text{RDM} & \longleftarrow (\text{Op}) \\ \text{RM} & \longleftarrow ((\text{RDM})) \\ \text{AC} & \longleftarrow \text{AC} + (\text{RM}) \end{array}$$

- **Carga:** se debe almacenar en el acumulador un dato contenido en memoria



$RDM \leftarrow (Op)$   
 $RM \leftarrow ((RDM))$   
 $AC \leftarrow (RM)$

- **Almacenamiento:** se debe “guardar” en memoria el contenido del acumulador

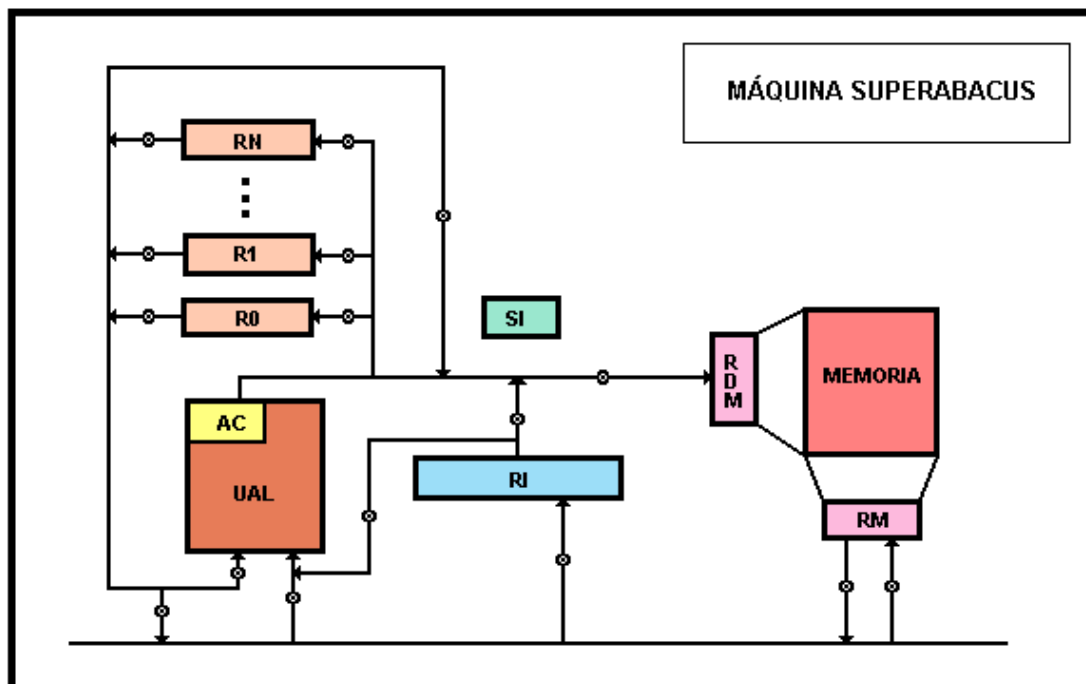
$RDM \leftarrow (Op)$   
 $RM \leftarrow (AC)$   
 $(RDM) \leftarrow (RM)$

- **Bifurcación:** se debe “saltar” a la dirección indicada en la instrucción. La dirección de bifurcación debe ser transferida al RPI (para buscar la próxima instrucción a ejecutar).

$RPI \leftarrow (Op)$

### SUPERABACUS

Superabacus es una maquina de tercera generación ya que posee un conjunto de registros banalizados, es decir, utilizables tanto como registros aritméticos o, según las condiciones de direccionamiento, como registro base o como registro de índice para cálculo de direcciones. Todos los cálculos se realizan en un solo sumador que actúa a la vez de unidad aritmético-lógica y de unidad de calculo de direcciones.



Posee sensiblemente características idénticas a Abacus, excepto en un punto: el ciclo de memoria equivale a cuatro impulsos de reloj, en lugar de dos, lo que autoriza entre otras

### 75.03 Organización del Computador

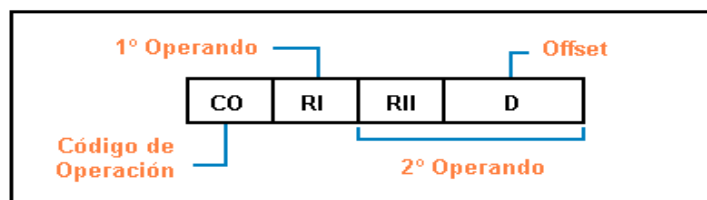
cosas a efectuar una modificación de dirección (sumar el contenido de un registro con la dirección indicada en la instrucción) sin retrasar por ello el inicio del ciclo de procesamiento.

Como características principales podemos enumerar las siguientes:

1. Posee un conjunto de registros generales éstos pueden contener datos o direcciones.
2. No tiene **RPI** se asigna esa función a **R0** (registro cero), cuyo incremento se consigue por transferencia vía el sumador.
3. Es una máquina de **dos direcciones** (1º y 2º operando).
4. La **UAL** se utiliza tanto para calcular direcciones como para operar con los datos.

#### Registro de Instrucción

En Superabacus la estructura del **RI** es diferente respecto de la Máquina Abacus; es un registro que alberga instrucciones de dos operandos. Posee Código de Operación, 1º Operando (RI) y 2º Operando (RII - D). La siguiente figura representa la estructura del **RI**:



RI y RII representan dos registros diferentes (se los indica con su número Ej.: R1, R5, etc.). El segundo operando puede contener la dirección de una celda de memoria, se obtiene sumando el Offset (desplazamiento) al contenido del registro indicado: **A (celda) = (RII) + D**.

#### Fase de búsqueda

Al igual que en el caso de la máquina Abacus, la fase de búsqueda de las instrucciones de Superabacus es común para todas.

RDM ← ( R0 )  
 RM ← ( ( RDM ) )  
 RI ← ( RM )  
 AC ← ( R0 )  
 R0 ← ( AC ) + 1

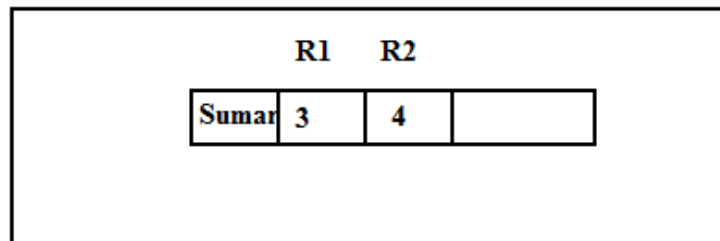
Como se puede apreciar, la gran diferencia con la máquina Abacus se da en la forma en que se incrementa el valor del R0 que hace las veces de RPI. En este caso ese incremento se resuelve usando la UAL en vez del SI.

#### Instrucciones

Las operaciones pueden ser entre registros, entre un registro y un dato inmediato o entre un registro y un operando en memoria. Analizaremos cada caso para una instrucción SUMAR.

1. **Sumar Registro:** se suma el contenido de ambos registros y el resultado se almacena en aquel que se indica en el primer operando.

**SUMAR RI, RII (Ejemplo SUMAR 3, 4)**



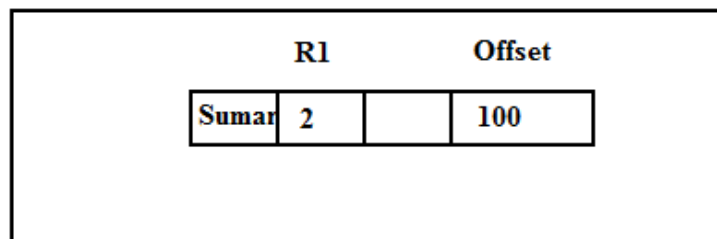
AC      ← ( R3 )

AC      ← ( R4 ) + ( AC )

R3      ← ( AC )

2. **Sumar Inmediato:** se suma al registro indicado en el 1º operando el dato inmediato almacenado en la instrucción. El resultado se almacena en el registro.

**SUMAR RI, DII (Ejemplo SUMAR 2, 100)**

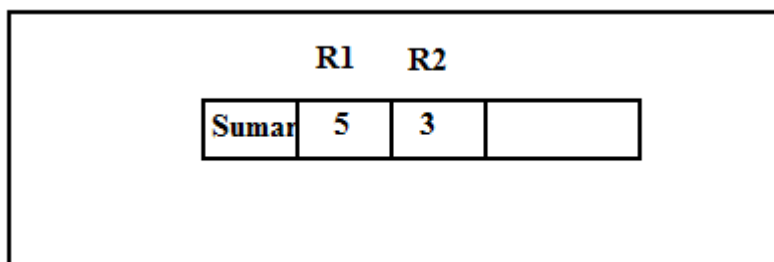


AC      ← ( R2 ) + 100

R2      ← ( AC )

3. **Sumar Palabra en memoria (Registro Indirecto):** se suma al registro indicado en el primer operando el contenido de la celda de memoria cuya dirección está dada por el contenido del registro indicado en el segundo operando. El resultado queda almacenado en el registro del primer operando:

**SUMAR RI, (RII) (Ejemplo SUMAR 5, (3) )**



AC  $\leftarrow$  ( R3 )  
 RDM  $\leftarrow$  ( AC )  
 RM  $\leftarrow$  ( ( RDM ) )  
 AC  $\leftarrow$  ( R5 ) + ( RM )  
 R5  $\leftarrow$  ( AC )

En este caso podemos notar que dado que el acumulador tiene una entrada desde los registros y una desde la memoria es posible realizar simultáneamente la suma del R5 y el RM.

4. **Sumar Palabra en memoria (Desplazamiento):** se suma al registro indicado en el primer operando el contenido de la celda de memoria cuya dirección está dada por el contenido del registro indicado en el segundo operando más el Offset. El resultado queda almacenado en el registro del primer operando:

**SUMAR RI, DI (RII) (Ejemplo SUMAR 5, 20 (3) )**

	R1	R2	Offset
Sumar	5	3	20

AC  $\leftarrow$  ( R3 ) + 20  
 RDM  $\leftarrow$  ( AC )  
 RM  $\leftarrow$  ( ( RDM ) )  
 AC  $\leftarrow$  ( R5 ) + ( RM )  
 R5  $\leftarrow$  ( AC )

En este caso podemos notar que dado que el acumulador tiene una entrada desde los registros y una desde la memoria es posible realizar simultáneamente la suma del R5 y el RM.

Mirando las instrucciones ejecutadas, nos preguntamos, ¿Como se da cuenta la maquina Superabacus en el llenado de los operandos? Y la respuesta es la siguiente, cada instrucción **Sumar** tiene un código de operación diferente en cada caso.