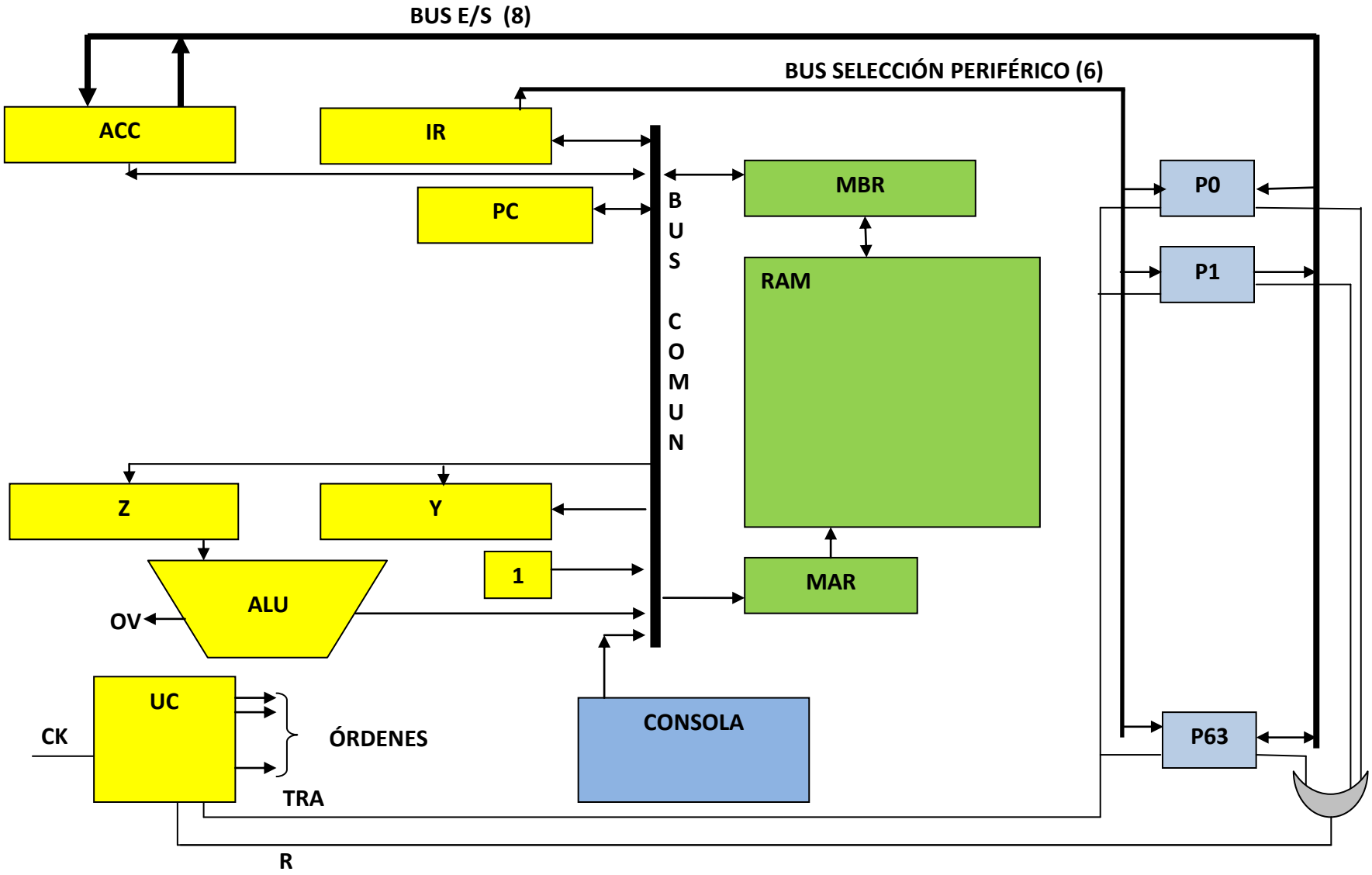
DESDE
LA MÁQUINA ELEMENTAL
A
LA MÁQUINA CONVENCIONAL

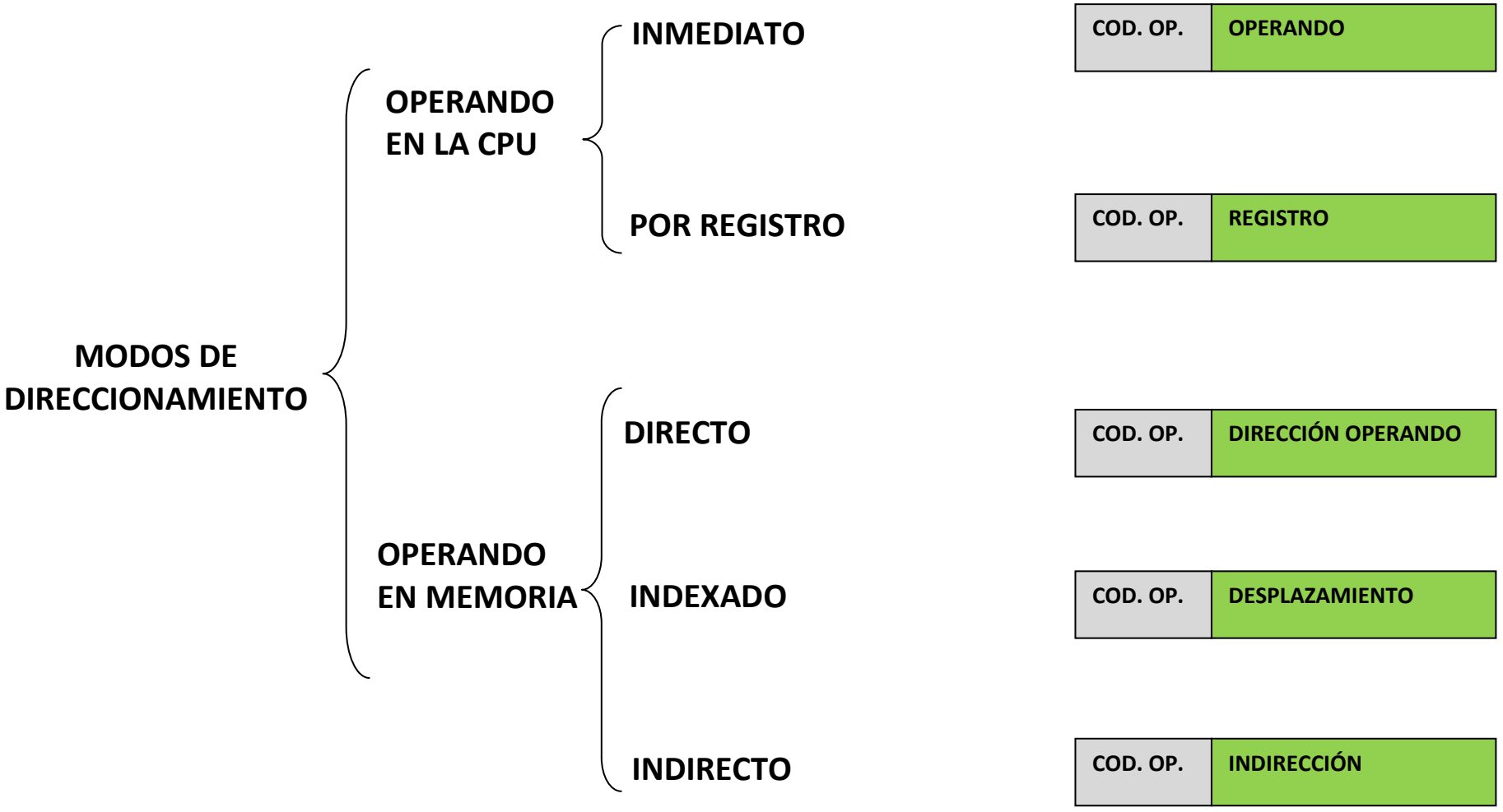
MÁQUINA ELEMENTAL



¿PORQUÉ MEJORAR LA MÁQUINA ELEMENTAL?

- **Facilitar la tarea del programador y el rendimiento del Sistema.**
- **Posibilitar:**
  - **La *modificación de direcciones*: agregar modos de direccionamiento: nuevos Registros ÍNDICE, nuevas instrucciones.**
  - **La programación de *bucles iterativos*: nuevos Registros ÍNDICE, nuevas instrucciones que implementan Contadores**
  - **Facilidad en la *reubicación de programas* residentes: nuevos Registros BASE**
  - **Manejo de E/S mediante *INTERRUPCIONES*: más hardware, más software.**

MODOS DE DIRECCIONAMIENTO



NUEVOS REGISTROS

RIF (16)	REGISTRO ÍNDICE FUENTE
RID (16)	REGISTRO ÍNDICE DESTINO
RB (16)	REGISTRO BASE

FORMATO DE INSTRUCCIÓN VARIABLE

COD. OP. (5)	MOD. (2)	DIRECCIÓN (16)

REGISTROS ÍNDICE I

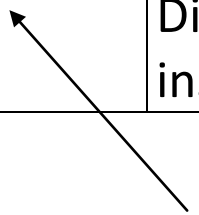
COD. OP. (5)	MOD. (2)	DIRECCIÓN (16)

MOD	M.D.	COMENTARIOS
00	DIRECTO	EL CAMPO DIRECCIÓN ES LA DIRECCIÓN DEL OPERANDO
01	INDIRECTO	EL CAMPO DIRECCIÓN ES LA DIRECCIÓN DE LA DIRECCIÓN DEL OPERANDO (INDIRECCIÓN)
10	INDEXADO RIF	EL CAMPO DIRECCIÓN + RIF ES LA DIRECCIÓN DEL OPERANDO
11	INDEXADO RID	EL CAMPO DIRECCIÓN + RID ES LA DIRECCIÓN DEL OPERANDO

REGISTROS ÍNDICE II

NUEVAS INSTRUCCIONES RELACIONADAS

RXT Reg, Reg	Copiar el contenido del Reg. al Reg.
SIX Reg, Dirección	Guardar Registro en la Dirección de Memoria
LIX Reg, Dirección	Cargar Registro desde la Dirección de Memoria
ENI Reg, Valor	Cargar Registro con la cantidad entera Valor
INI Reg, Valor	Incrementar Registro con la cantidad entera Valor
IJP Reg, Dirección	Si Reg. No es cero, decrementar Reg. y saltar a Dirección. Si Reg. es cero seguir con la próxima instrucción



PERMITE IMPLEMENTAR UN CONTADOR PARA BUCLE ITERATIVO

REGISTRO BASE

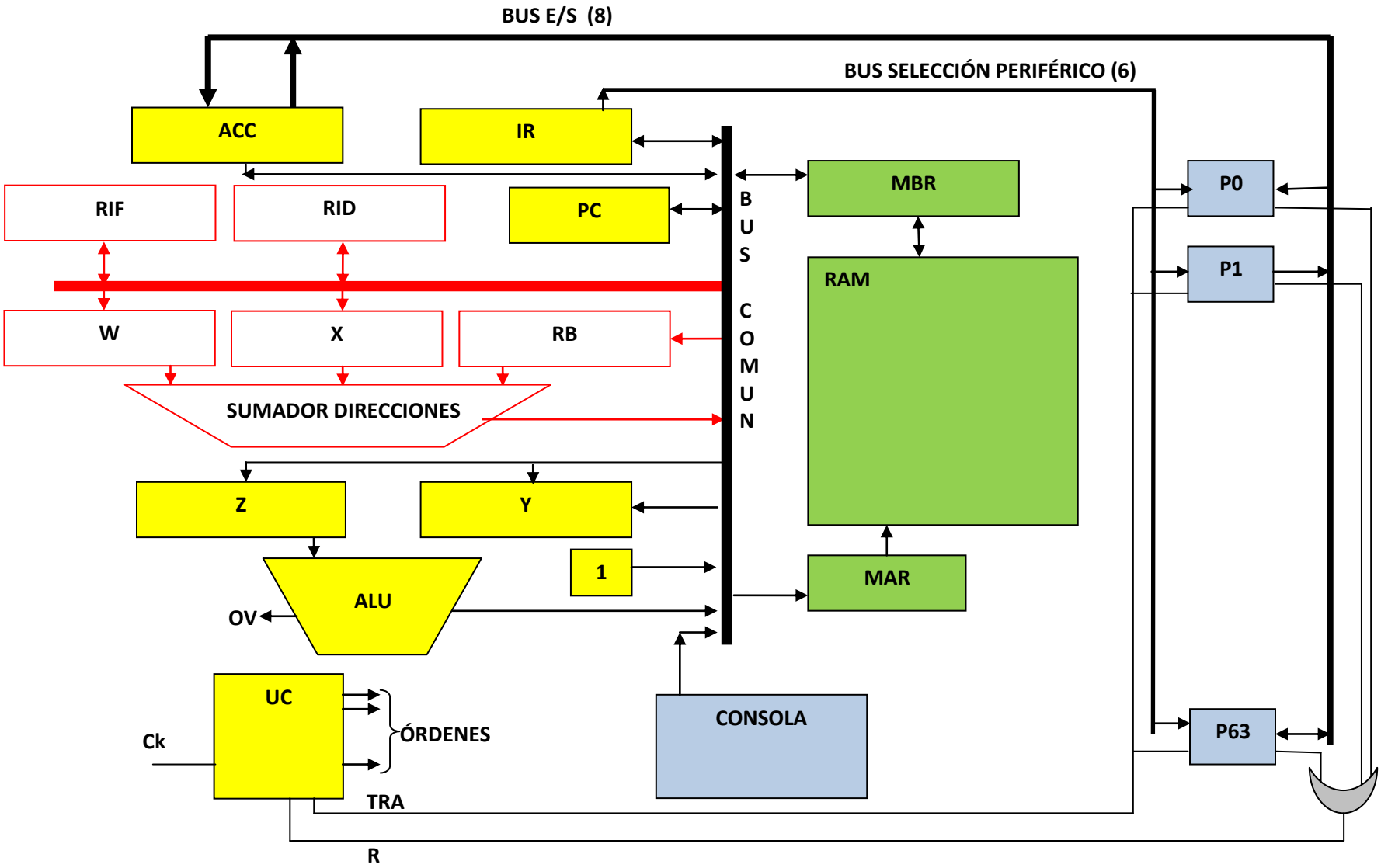
RB (16)

- **LA UNIDAD DE CONTROL SUMA EL CONTENIDO DEL RB AL CAMPO DE DIRECCIÓN DE LA INSTRUCCIÓN (CUANDO ES UNA DIRECCIÓN), EN TIEMPO DE EJECUCIÓN**
- **EL CONTROL DEL REGISTRO RB ESTÁ RESERVADO AL ADMINISTRADOR.**

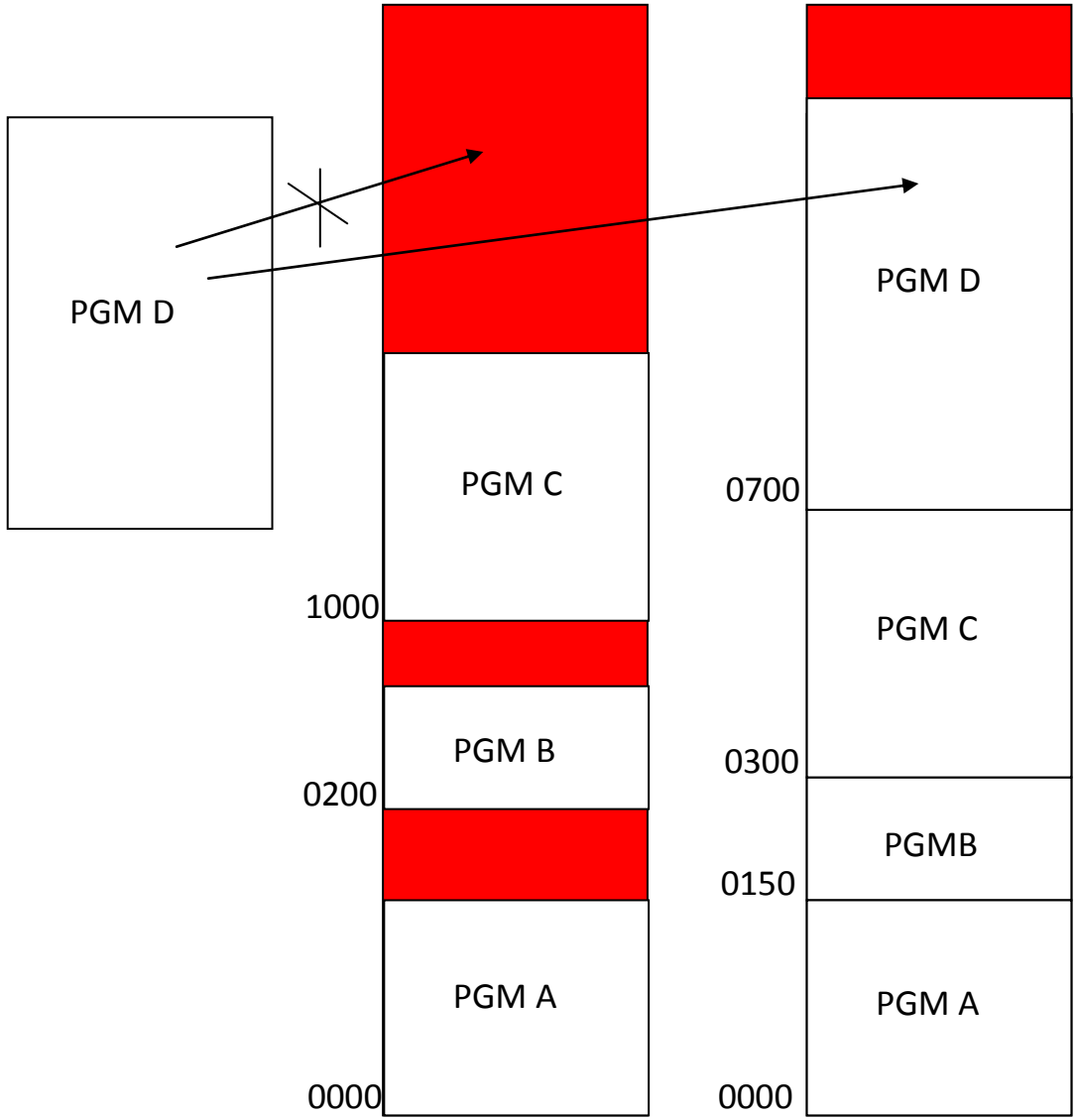
NUEVAS INSTRUCCIONES RELACIONADAS

ENI RB	***XXXXXX***	**Carga al Registro Base (RB) con el número entero *XXXXXX*. Solo debe ser usada por el administrador del Sistema. (Instrucción privada).**

MÁQUINA CONVENCIONAL



REUBICACIÓN DE PROGRAMAS RESIDENTES I



REUBICACIÓN DE PROGRAMAS RESIDENTES II

La existencia de más de un programa residente en memoria listo para ser ejecutado se llama MULTIPROGRAMACIÓN. Esto genera la necesidad de llenar la memoria evitando espacios vacíos, es decir: La Reubicación de programas.

Todos los programadores podrían escribir sus programas comenzando en la posición cero, cuando se carga en Memoria, llamado ENSAMBLADO, (o se reubica) será entonces necesario:

A) CORRECCIÓN DE DIRECCIONES
B) REUBICACIÓN DEL PROGRAMA

A) CORRECCIÓN DE DIRECCIONES

Ejemplo:

```
PGM X
00   LDA 06
01   NOT
02   AND 07
03   JMA 04
04   STA 10
05   HLT
06       Dato1
07       Dato2
10       Resultado
```

Cargando el RB con el valor de dirección inicial donde se ensamblará el PGM se soluciona la *corrección de direcciones* ya que su contenido será sumado automáticamente a las direcciones en rojo

B) REUBICACIÓN DEL PGM

Es necesario reubicar el PGM B desde su ubicación actual (dirección inicial 200) a la nueva ubicación (dirección 150):

- **PGM de reubicación en la Máquina Elemental:**

```
LDA 200
STA 150
LDA 201
STA 151
LDA 202
STA 152
-----
-----
```

- **PGM de reubicación en la Máquina mejorada (BUCLE ITERATIVO)**

```
        ENI A, Long. Bloque
        ENI RIF, 0
BUCLE   LDA2, 200
        STA2, 150
        INI RIF, 1
        IJP A, BUCLE
```

- _____

OTROS EJEMPLOS

PROCESAMIENTO DE BLOQUES DE DATOS

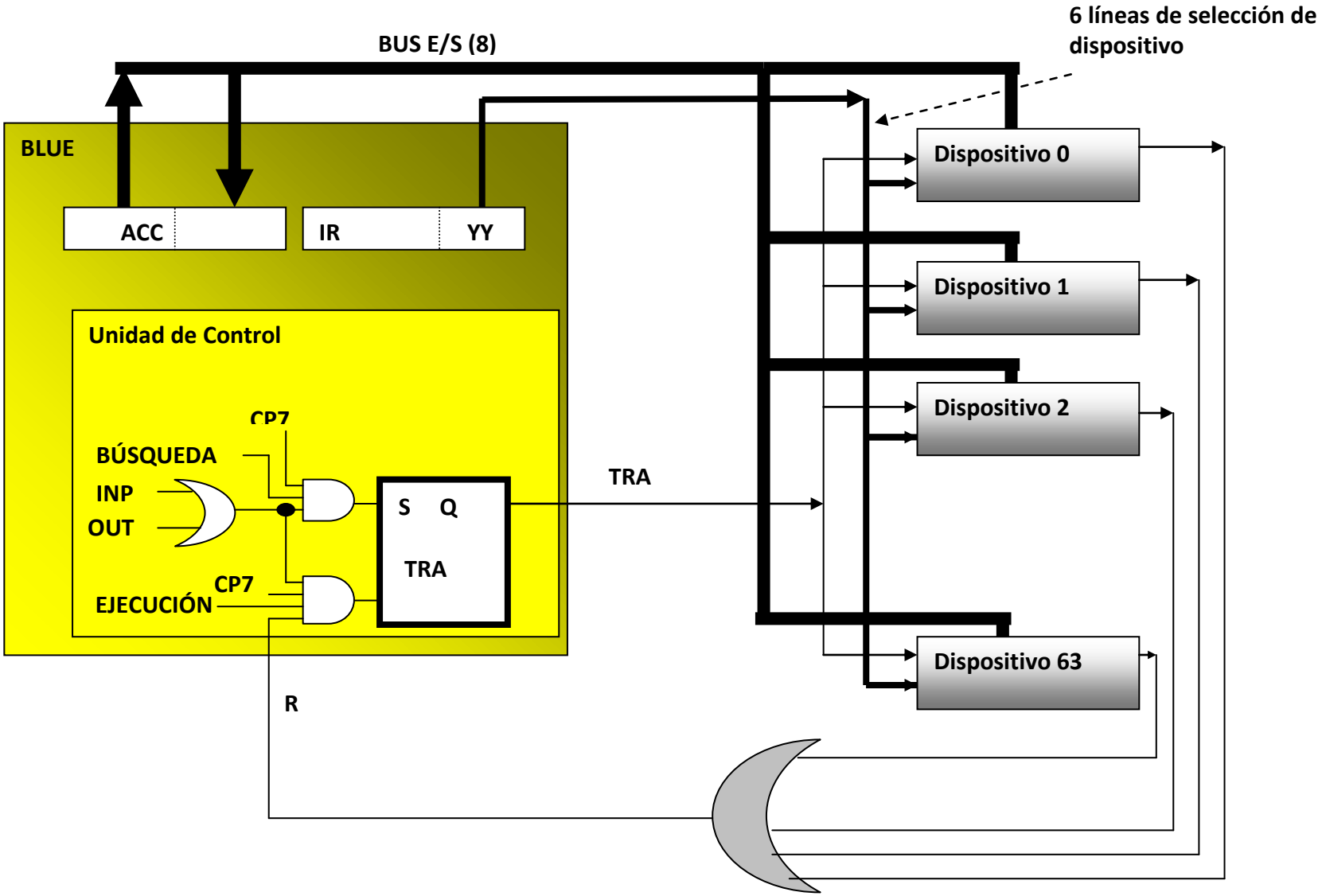
Supongamos que deseamos sumar sucesivamente un conjunto de 100 números ubicados correlativamente en la memoria.
Deberíamos:

- **Tomar uno a uno los números**
- **Sumarlos**
- **Colocar la suma parcial en alguna posición de memoria hasta llegar al número 100**

SUBRUTINAS REENTRANTES

Supongamos que deseamos escribir una subrutina que multiplique dos números. Esta subrutina debe estar disponible para cualquier programa, el área de memoria que utiliza para los datos debe corresponderse con el programa que la llamó y, cuando sea cargada en memoria, residirá en un área que no se puede prever con anticipación.

ENTRADA/SALIDA EN LA MÁQUINA ELEMENTAL



INTERRUPCIONES

“Agregar hardware de tal forma que cuando eventos externos requieren atención, se produzca una suspensión (*interrupción*) automática del programa corriente y se transfiera el control temporariamente a una rutina diseñada especialmente para manejar estos eventos. Finalmente se retorna al programa corriente como si nada hubiera pasado...”. Caxton Foster

INTERRUPCIONES

Es un Proceso que involucra Hardware y Software a fin de resolver eficazmente las transferencias de entrada/salida evitando que la CPU pierda tiempo esperando al Periférico y además resuelva el problema de eventos externos ex temporáneos.

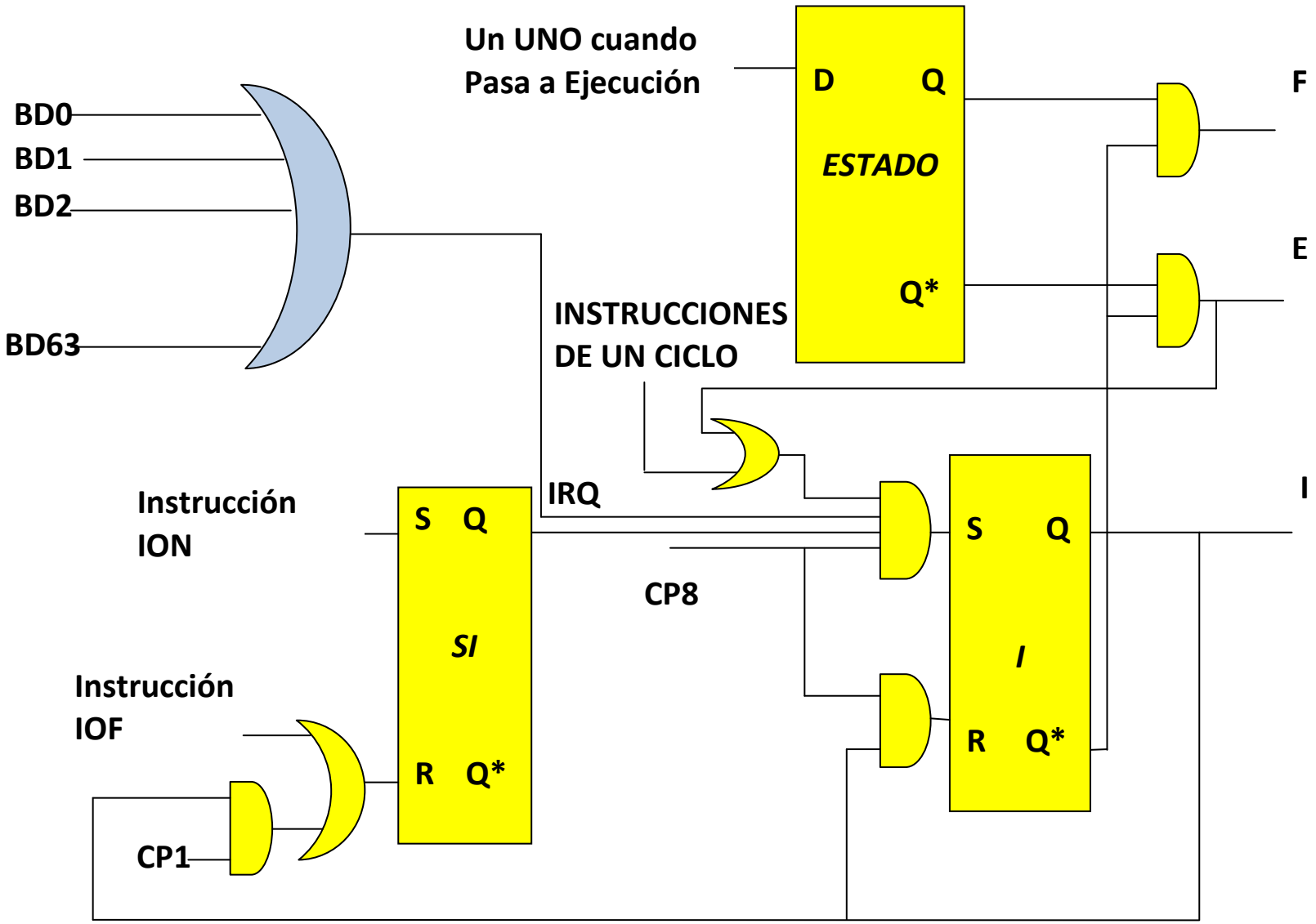
Los periféricos deben ser capaces de solicitar atención.

La UC debe disponer de hardware para chequear permanentemente si algún periférico solicita atención.

Cuando una solicitud de Interrupción ocurre, la UC debería:

- **Terminar de ejecutar la instrucción corriente.**
- **Inhabilitar el Sistema de Interrupciones.**
- **Salvar la dirección de retorno (Eventualmente salvar CONTEXTO)**
- **Saltar a un programa que atienda al periférico (RUTINA de INTERRUPCIÓN)**
- **Terminada la Transferencia restaurar CONTEXTO, habilitar el Sistema de Interrupciones y volver al PGM corriente.**

SISTEMA ELEMENTAL DE INTERRUPCIONES I

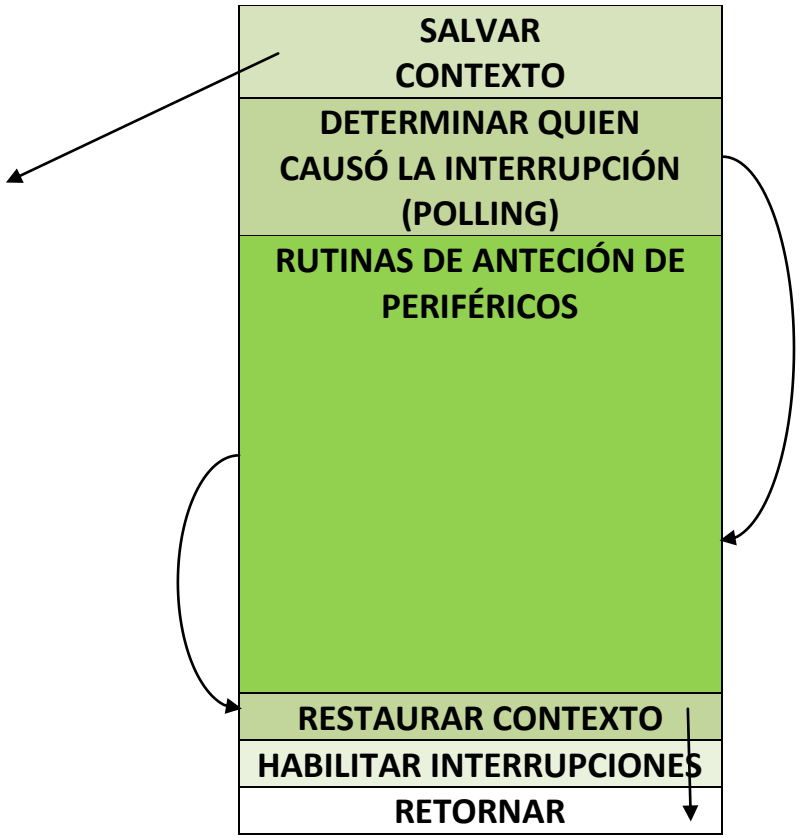


SISTEMA ELEMENTAL DE INTERRUPCIONES II

CICLO DE INTERRUPCIÓN

Pulso	Acción
CP1	Biestable SI = 0
CP2	Envíe PC, Cargue MBR
CP3	Envíe 0, Cargue MAR, OE
CP4	Envíe 1, Cargue PC
CP5	-
CP6	-
CP7	-
CP8	Biestable I = 0

RUTINA DE INTERUPCIÓN



SISTEMA ELEMENTAL DE INTERRUPCIONES III

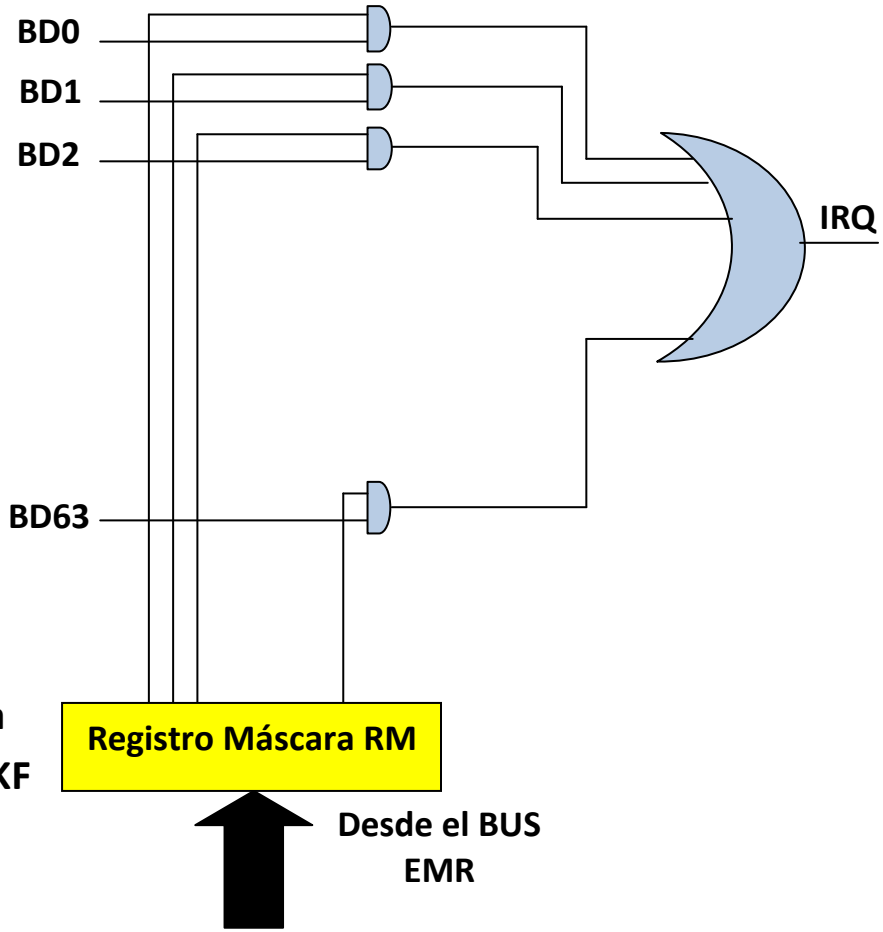
DETERMINACIÓN DE QUIÉN CAUSÓ LA INTERRUPCIÓN POR SOFTWARE (POLLING)

Agregar la instrucción SKF BDn: Omite la próxima instrucción si BDn = 0

PGM POLLING

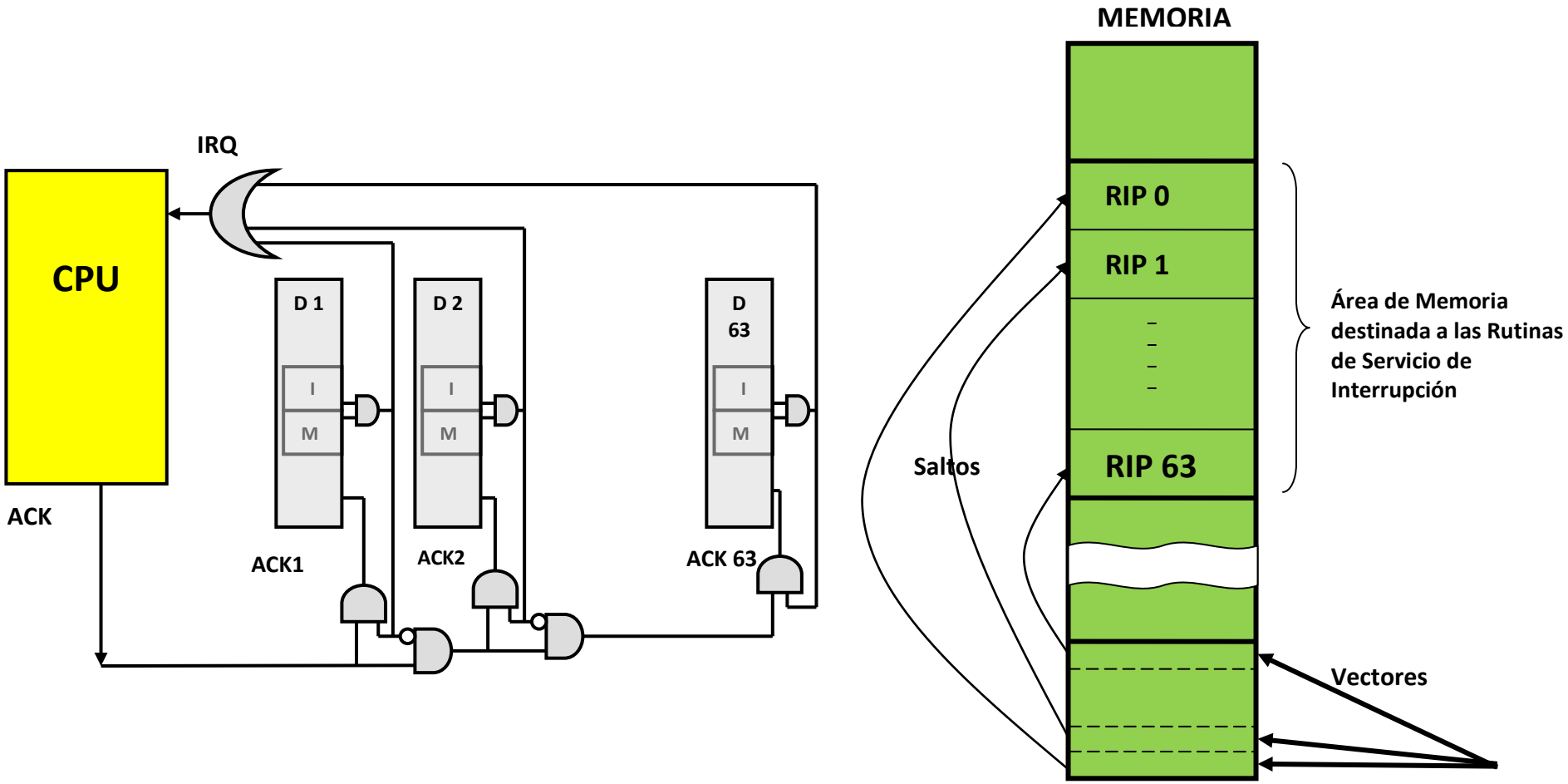
SKF BD0
JUMP SRD0
SKF BD1
JUMP SRD1
-
-
-
SKF BD62
JUMP SRD62
JUMP SRD63

- **Las Banderas de Dispositivo (BDn) son puestas a 1 por el periférico cuando solicita atención, y puestas a 0 por la instrucción SKF**
- **La prioridad es por software**



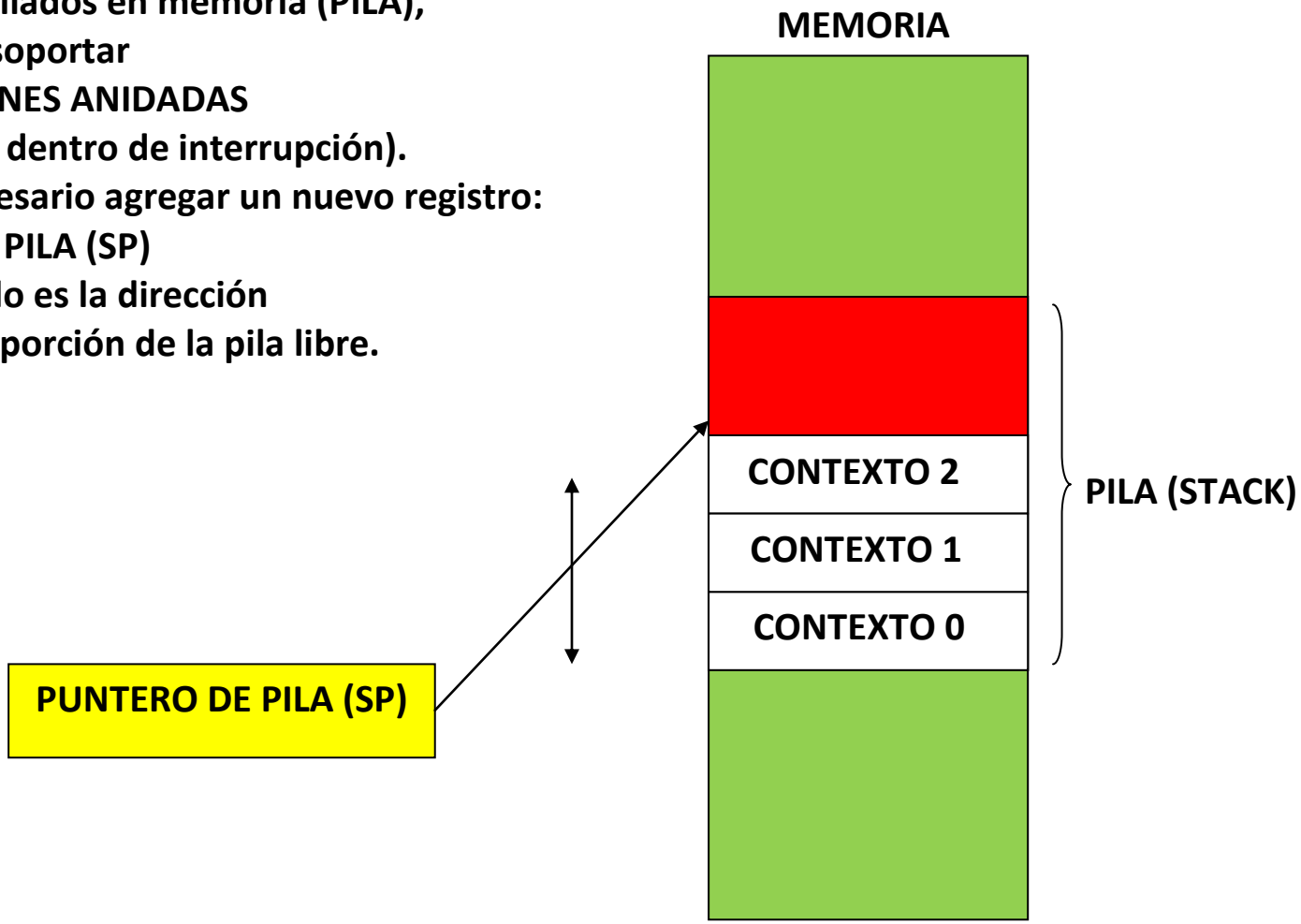
SISTEMA ELEMENTAL DE INTERRUPCIONES IV

DETERMINACIÓN DE QUIÉN CAUSÓ LA INTERRUPCIÓN MÉTODO VECTORIZADO



SISTEMA ELEMENTAL DE INTERRUPCIONES V

Si los contextos (incluido el PC) se fueran guardando apilados en memoria (PILA), sería posible soportar INTERRUPCIONES ANIDADAS (Interrupción dentro de interrupción). Esto hace necesario agregar un nuevo registro: PUNTERO DE PILA (SP) cuyo contenido es la dirección de la primera porción de la pila libre.



CONCLUSIONES

- **Se han introducido algunas modificaciones a la MAQUINA ELEMENTAL para facilitar su programación y rendimiento:**
  - **INDEXACIÓN**
  - **DIRECCIONAMIENTO INDIRECTO**
  - **REGISTRO BASE**
  - **INTERRUPCIONES**
- **Adicionalmente surge como necesidad contar con un programa que administre eficazmente los recursos del Sistema: PROGRAMA MONITOR.**
  - **ADMINISTRAR LA MEMORIA**
  - **ADMINISTRAR E/S**
  - **ADMINISTRAR LA CPU**