
Estructura de Computadores

El computador simple

Autores: David Guerrero. Isabel Gómez. Alberto Molina.

Usted es libre de copiar, distribuir y comunicar públicamente la obra y de hacer obras derivadas siempre que se cite la fuente y se respeten las condiciones de la licencia Attribution-Share alike de Creative Commons.

Texto completo de la licencia: <http://creativecommons.org/licenses/by-nc-sa/3.0/es/>



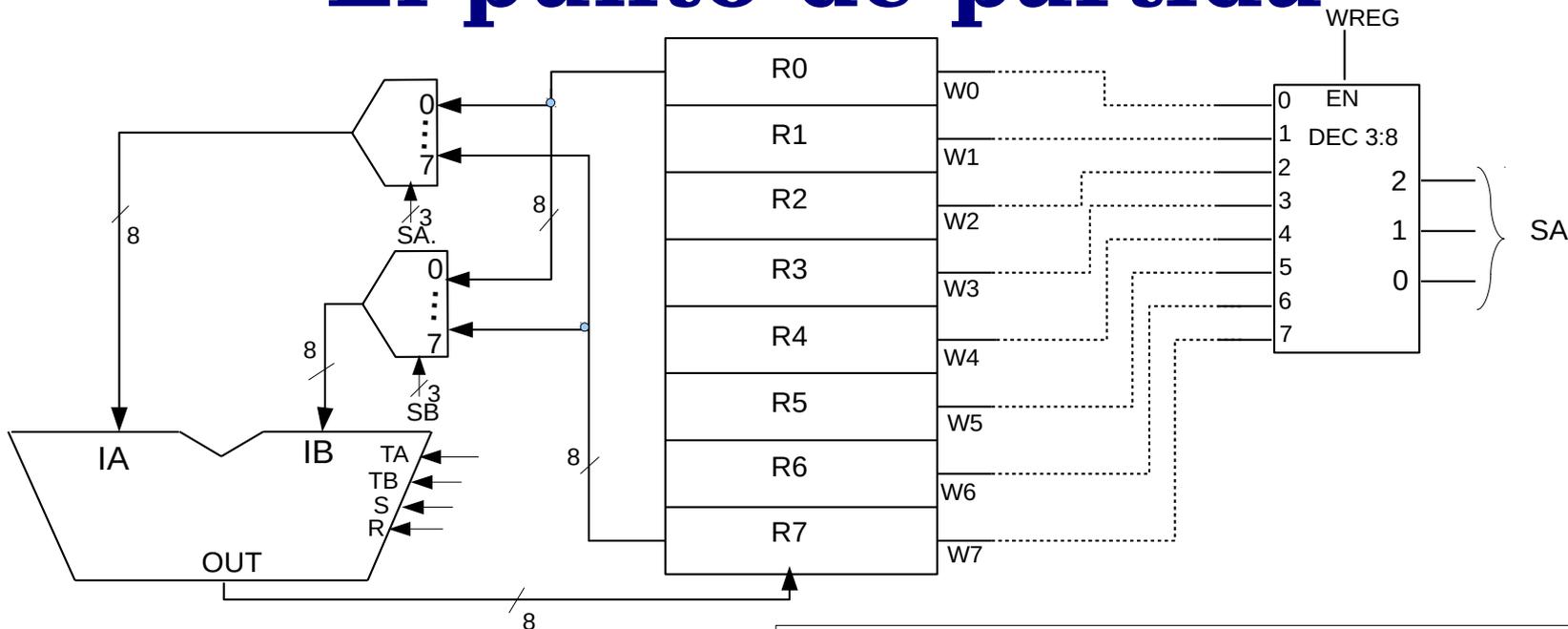
Guión

- ▶ **El punto de partida: La calculadora**
- ▶ **Automatización en la ejecución y almacenamiento de programa (CS1)**
- ▶ **Almacenamiento de los datos y ampliación de modos de direccionamiento (CS2)**
- ▶ **Diversificación de instrucciones. Ejecución no secuencial (CS3)**

El punto de partida

- ▶ Partimos de la calculadora planteada en el tema anterior que es un sistema en un único paso.
- ▶ La calculadora ejecuta cualquier posibilidad de suma o resta entre sus registros así como el movimiento de datos entre los mismos.
- ▶ Las operaciones se realizan en un único ciclo de reloj.

El punto de partida



Los registros fuente y destino pueden ser Cualquiera de los 8 pertenecientes a la arquitectura

TA TB R S	OUT
0 0 0 1	A+B
0 0 1 0	A-B
0 1 0 0	B
1 0 0 0	A



IR ₇ IR ₆	Operaciones
00	$R(SA) \leftarrow R(SA) + R(SB)$
10	$R(SA) \leftarrow R(SA) - R(SB)$
01	$R(SA) \leftarrow R(SB)$

Registro adicional que no posee la calc.

Es donde se guarda la información que pone el usuario

desde el exterior. Información sobre la operación a realizar

Y los datos implicados. El registro destino viene determinado por

El valor de los 3 bits SA y el fuente por el de los 3 SB.

El punto de partida

Se propone la realización de un instrucción más compleja: $R0 \leftarrow 3R1 - R2$

Secuencia de instrucciones a nivel ISP:

Instrucción 1. $R0 \leftarrow R1$

Instrucción 2. $R0 \leftarrow R0 - R2$

Instrucción 3. $R0 \leftarrow R0 + R1$

Instrucción 4. $R0 \leftarrow R0 + R1$

Las instrucciones

deben ser las de la calculadora.
Resta siempre en el mismo sentido
y sólo registros destino y fuente
implicados.

En este ejemplo se actualizaría 4 veces
el registro IR

Ventajas:

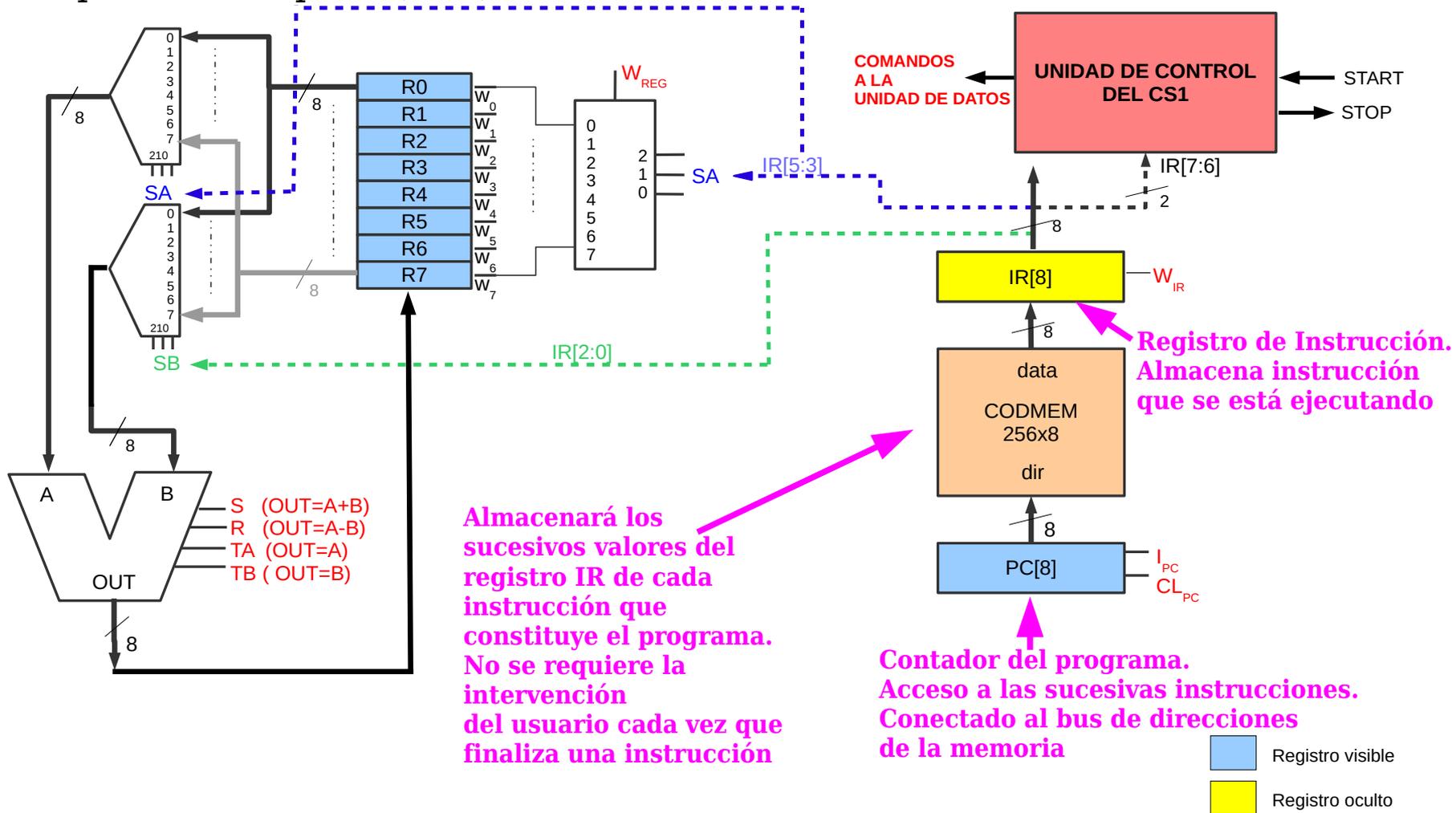
- Podemos resolver problemas más complejos que los resueltos por las instrucciones primitivas mediante una secuencia adecuada de las mismas.
- El usuario del sistema no necesita ser especialista en la electrónica del sistema.

Deficiencias:

- No hay AUTOMATIZACIÓN EN LA EJECUCIÓN del programa: para ejecutar cada instrucción el usuario debe proporcionar el valor de xs y esperar fin para cada una de las cuatro instrucciones.
- No hay PROGRAMA ALMACENADO: cada vez que se ejecuta una instrucción el usuario debe suministrar la siguiente.

Automatización en la ejecución

El Computador Simple 1 (CS1)



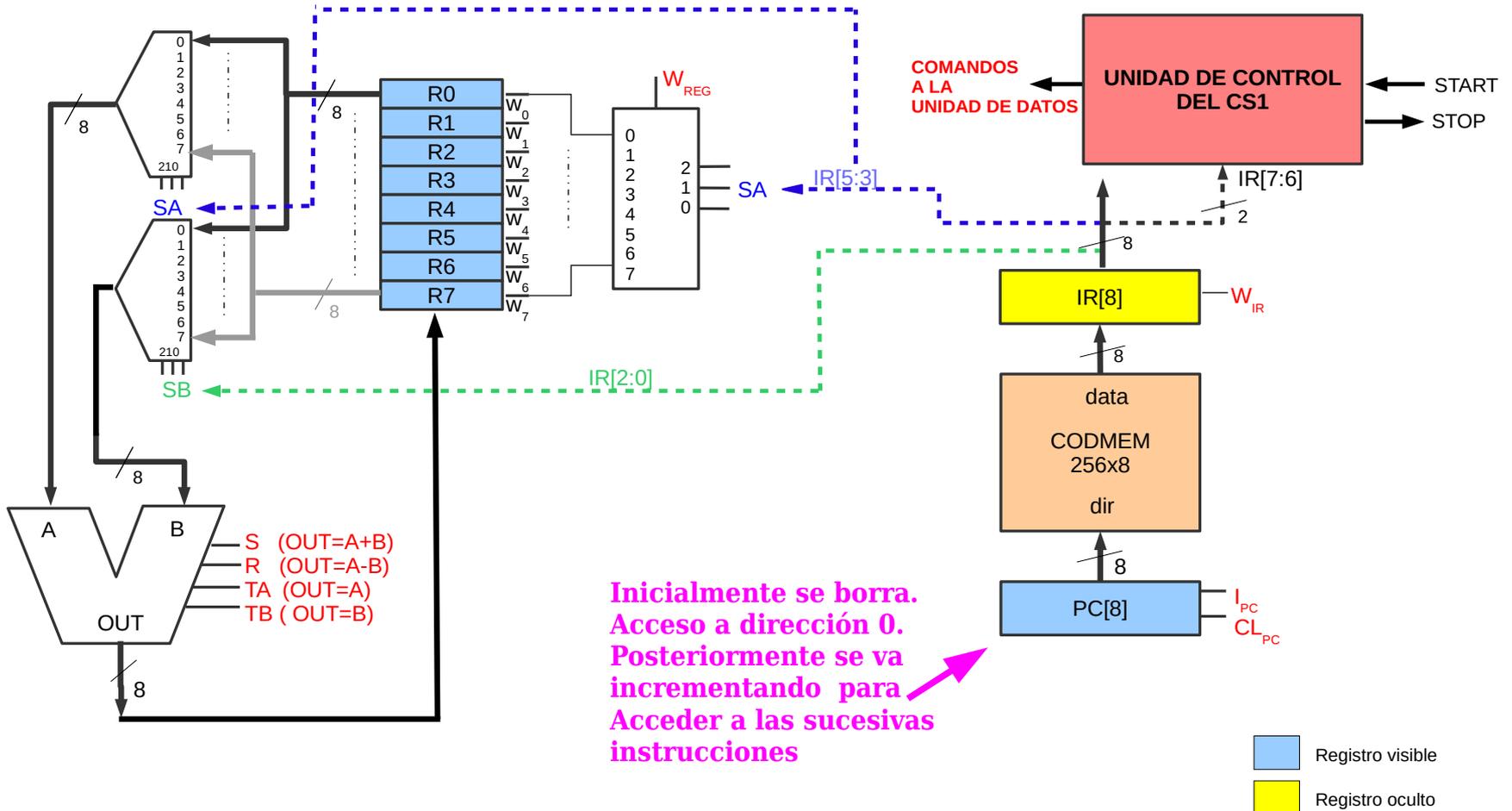
Almacenará los sucesivos valores del registro IR de cada instrucción que constituye el programa. No se requiere la intervención del usuario cada vez que finaliza una instrucción

Contador del programa. Acceso a las sucesivas instrucciones. Conectado al bus de direcciones de la memoria

Registro de Instrucción. Almacena instrucción que se está ejecutando

Automatización en la ejecución

El Computador Simple 1 (CS1)



Inicialmente se borra.
 Acceso a dirección 0.
 Posteriormente se va incrementando para
 Acceder a las sucesivas instrucciones

Automatización en la ejecución

- ▶ En el CS1 todas las instrucciones son de una palabra. Esto es una simplificación ya que en muchos sistemas reales son de longitud variable.
- ▶ Dada la simplicidad de la arquitectura, los programas siempre están almacenados a partir de la dirección 0 de la memoria y la ejecución es lineal.
- ▶ Instrucción en código máquina. Es el patrón de bits correspondiente a una instrucción.
- ▶ Formato de instrucción: indica como debe ser interpretada una instrucción en código máquina (código de operación y operandos).

Automatización en la ejecución

► Formato de instrucción del CS1

7	6	5	4	3	2	1	0
código de operación		registro destino			registro fuente		

En el CS1 los datos siempre están almacenados en sus 8 registros y por tanto los operandos están codificados por 6 bits. Los tres primeros identifican al registro destino y los tres últimos al fuente.

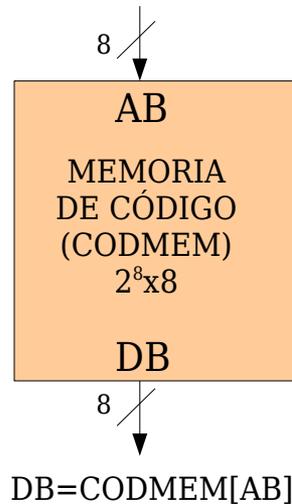
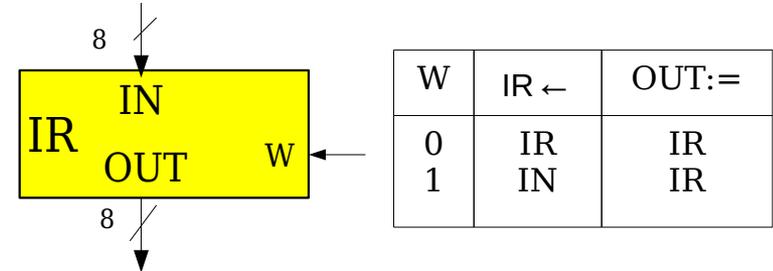
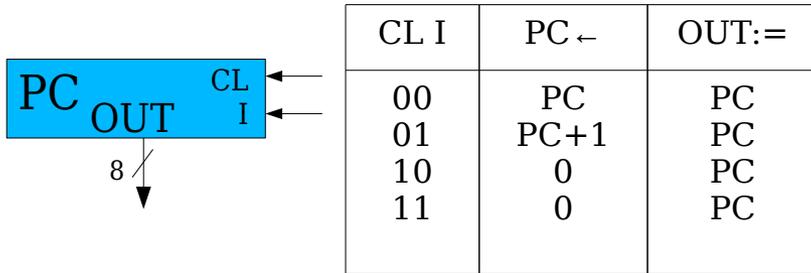
► Instrucciones. Se definen 4 instrucciones ya que se dispone de dos bits de código de operación.

CO: IR7 IR6	SINTAXIS	FUNCIÓN
00	ADD Rd,Rf	$Rd \leftarrow Rd + Rf$
10	SUB Rd,Rf	$Rd \leftarrow Rd - Rf$
01	MOV Rd,Rf	$Rd \leftarrow Rf$
11	STOP	NOP

En la sintaxis se han utilizado nemónicos que facilitan La tarea al programador.

Automatización en la ejecución

Descripción RT de los nuevos componentes



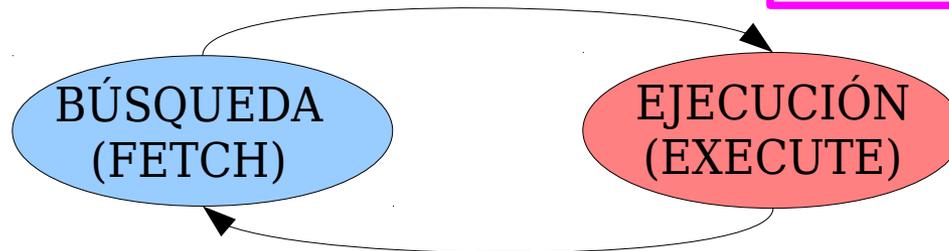
$$DB = \text{CODMEM}(AB)$$

La memoria siempre muestra el dato que es apuntado por la dirección.
Memoria de sólo lectura.

Automatización en la ejecución

► Diseño de la unidad de control: permite la ejecución automática del programa almacenado en la memoria.

► Ciclo de instrucción



La búsqueda consiste en el acceso a la MEMCOD
A la dirección apuntada por PC y la escritura del dato
Accedido en el registro IR

► La carta ASM del CS1 implementa los ciclos de todas las instrucciones del sistema.

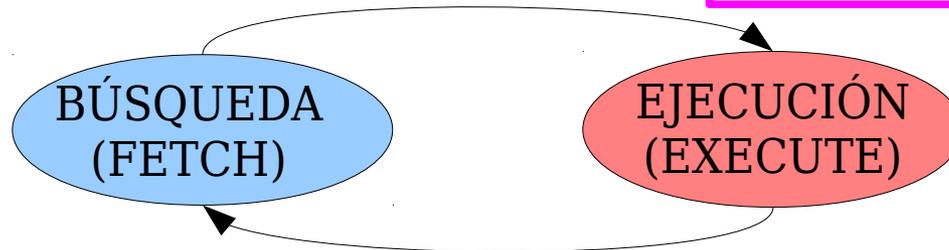
C.O: IR7 IR6	SINTAXIS	FUNCIÓN
00	ADD Rd,Rf	Rd ← Rd+Rf
10	SUB Rd,Rf	Rd ← Rd-Rf
01	MOV Rd,Rf	Rd ← Rf
11	STOP	NOP

Automatización en la ejecución

► Diseño de la unidad de control: permite la ejecución automática del programa almacenado en la memoria.

► Ciclo de instrucción

La ejecución consiste en que la unidad de control en base a la información de los bits del código de operación secuencia las transferencias entre registros de los que se compone una instrucción



► La carta ASM del CS1 implementa los ciclos de todas las instrucciones del sistema.

C.O: IR7 IR6	SINTAXIS	FUNCIÓN
00	ADD Rd,Rf	$Rd \leftarrow Rd+Rf$
10	SUB Rd,Rf	$Rd \leftarrow Rd-Rf$
01	MOV Rd,Rf	$Rd \leftarrow Rf$
11	STOP	NOP

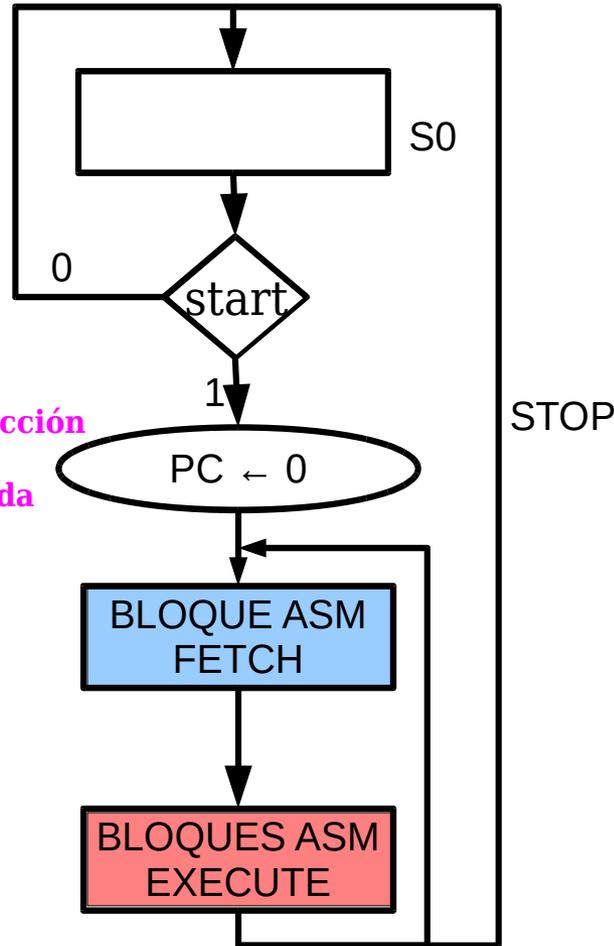
Los bits IR7 e IR6 son entradas a la Unidad de Control

Automatización en la ejecución

Carta ASM del CS1

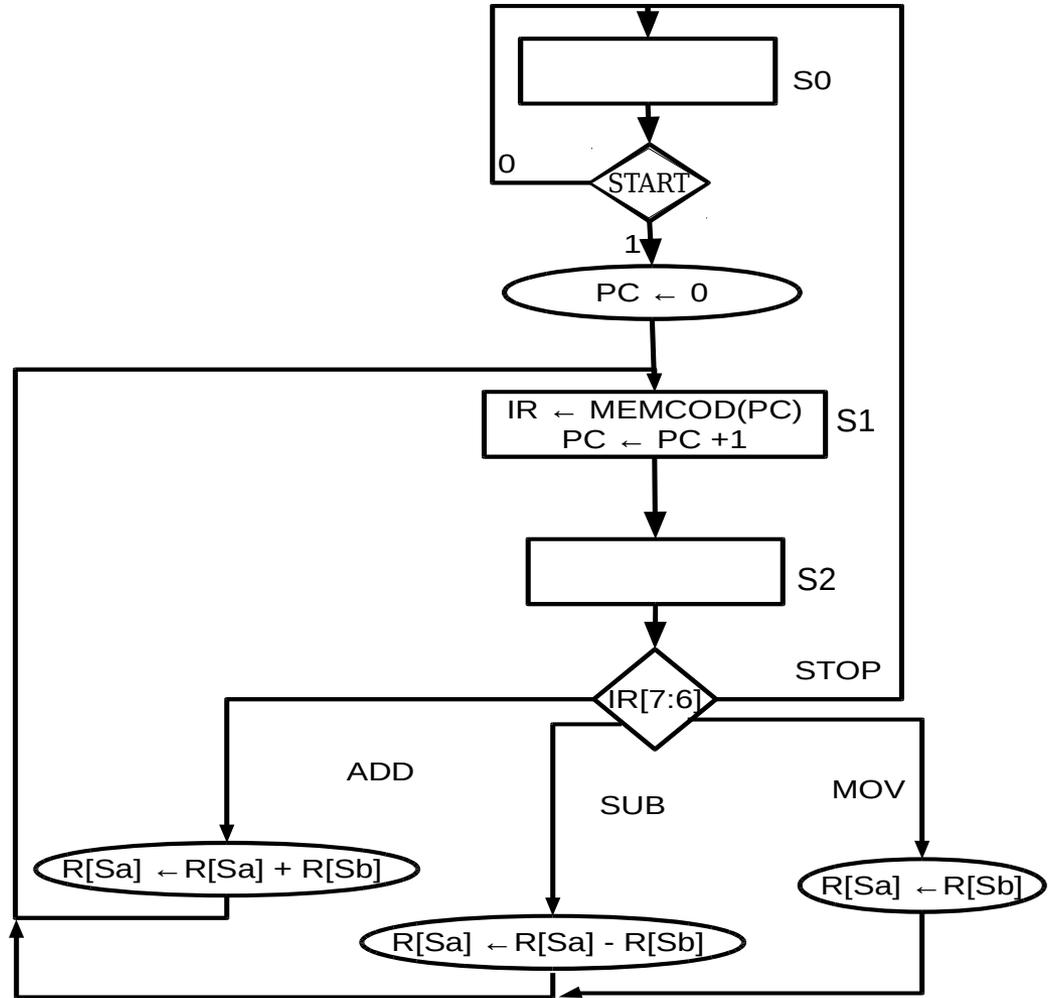
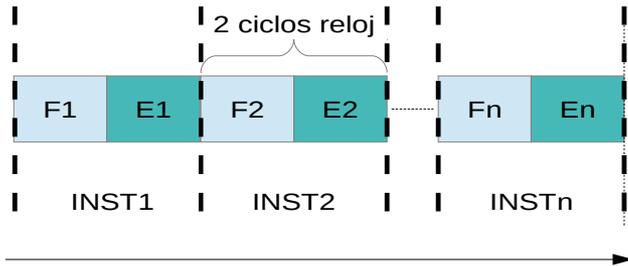
Esquema global:

1. Al comenzar la operación se borra PC
2. A continuación ciclo de búsqueda, la instrucción se guarda en IR.
3. El ciclo de ejecución será diferente para cada Instrucción.



Automatización en la ejecución

Carta ASM del CS1

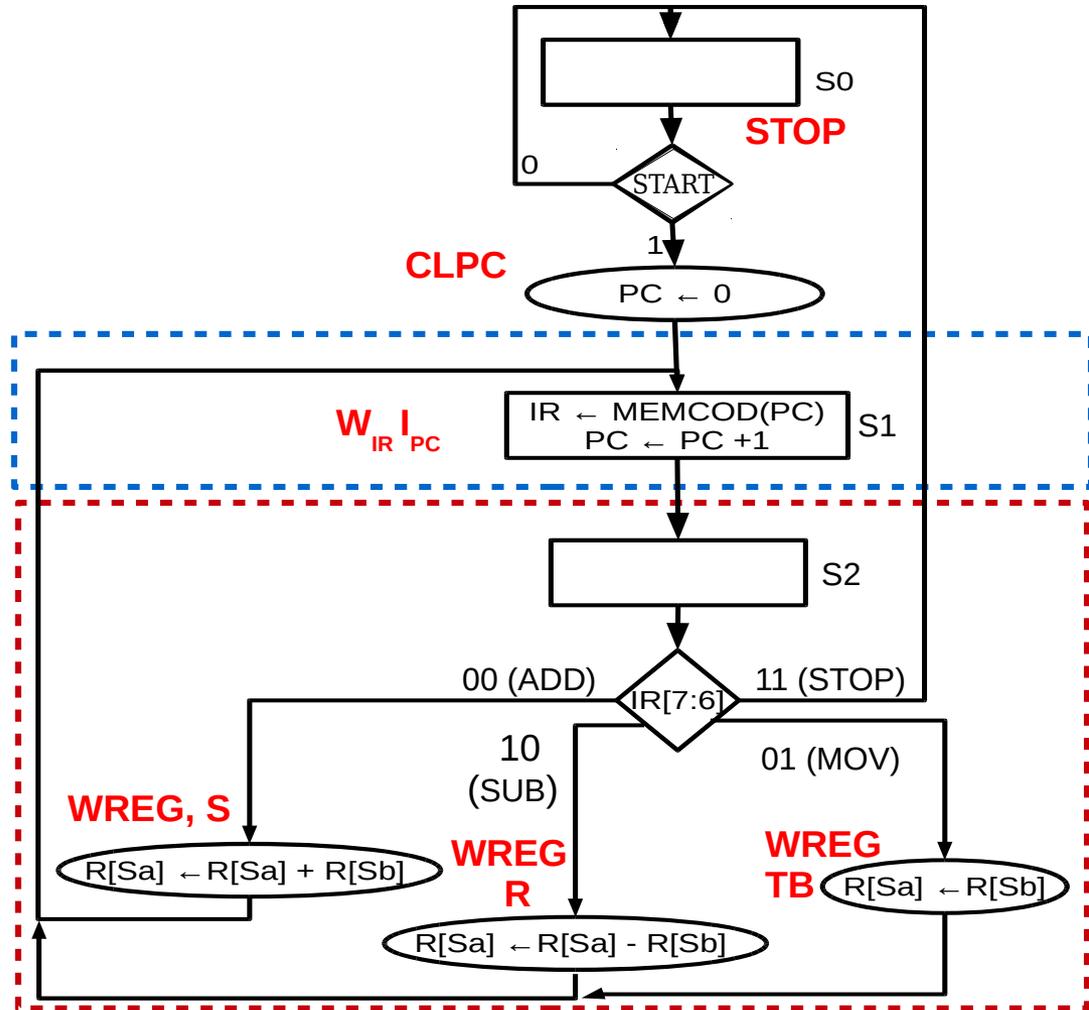


Automatización en la ejecución

Carta ASM del CS1

Estados de la fase de Búsqueda: S1.
Estados de la fase de Ejecución: S2

Se ha dibujado la carta ASM con las transferencias entre registros. Las señales que debe activar la unidad de control para que se produzcan dichas transferencias se encuentran fuera de la carta



Automatización en la ejecución

Ejemplo de uso del CS1: Escribir un programa que realice la siguiente operación: $R6 \leftarrow 3R4 - 2R1$

Programa

```
MOV R6,R4  
SUB R6,R1  
ADD R6,R6  
ADD R6,R4  
STOP
```

\$Posición	contenido
\$00	01 110 100
\$01	10 110 001
\$02	00 110 110
\$03	00 110 100
\$04	11 - - - - -

MEMORIA DE CÓDIGO