
Índice

1. Limitaciones de la calculadora simple

2. El Computador Simple 1 (CS1)

(concepto de Programa almacenado en memoria)

3. El Computador Simple 2 (CS2)

(memoria de datos y memoria de programa)

4. El Computador Simple CS2010

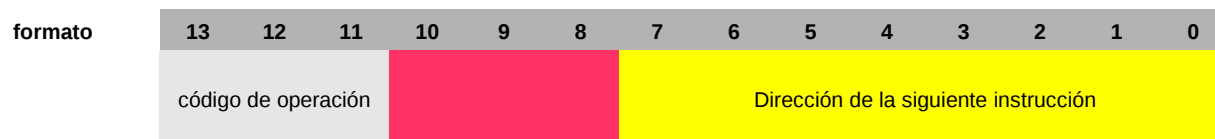
(ampliación del conjunto de instrucciones)

Limitaciones del computador CS2

- El computador simple 2 presenta muchas limitaciones:
 - **Imposibilidad de realizar saltos** en la ejecución del programa.
 - **Ausencia de variables de estado que informen del resultado de las operaciones.**
 - No permite direccionamiento inmediato.
 - No puede interactuar con el mundo exterior.
- **Se propone una nueva arquitectura** pensada para solventar estas deficiencias: **el CS2010.**
- CS2010 es un computador “académico”, pero “compatible” en juego de instrucciones con un microcontrolador real (AVR)

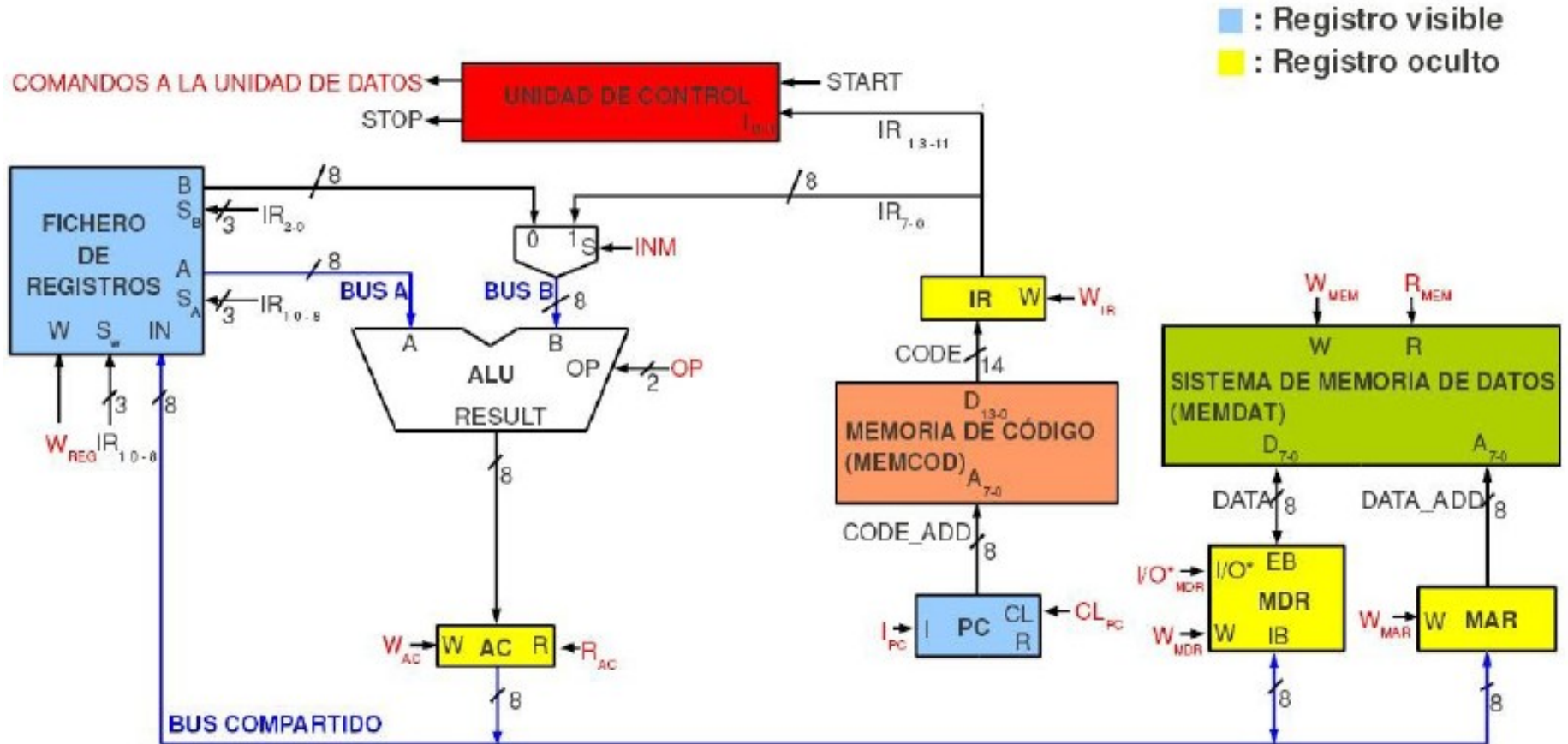
Instrucciones de salto

- El CS2 no incorpora ninguna instrucción de salto
- Existen varios tipos de saltos: **incondicionales** y **condicionales**
- Ambos requieren poder **modificar el contenido del PC**
- La instrucción de salto debe contener **la dirección** de memoria donde se encuentra la siguiente instrucción a ejecutar



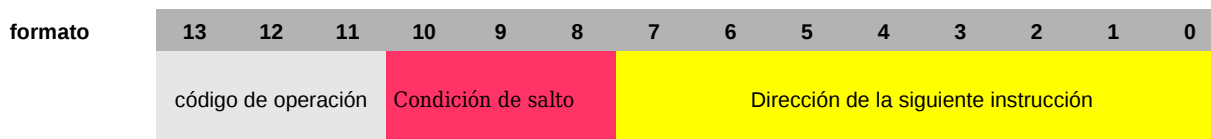
Arquitectura del CS2

Adaptación para salto incondicional
(A completar por el alumno)



Instrucciones de salto

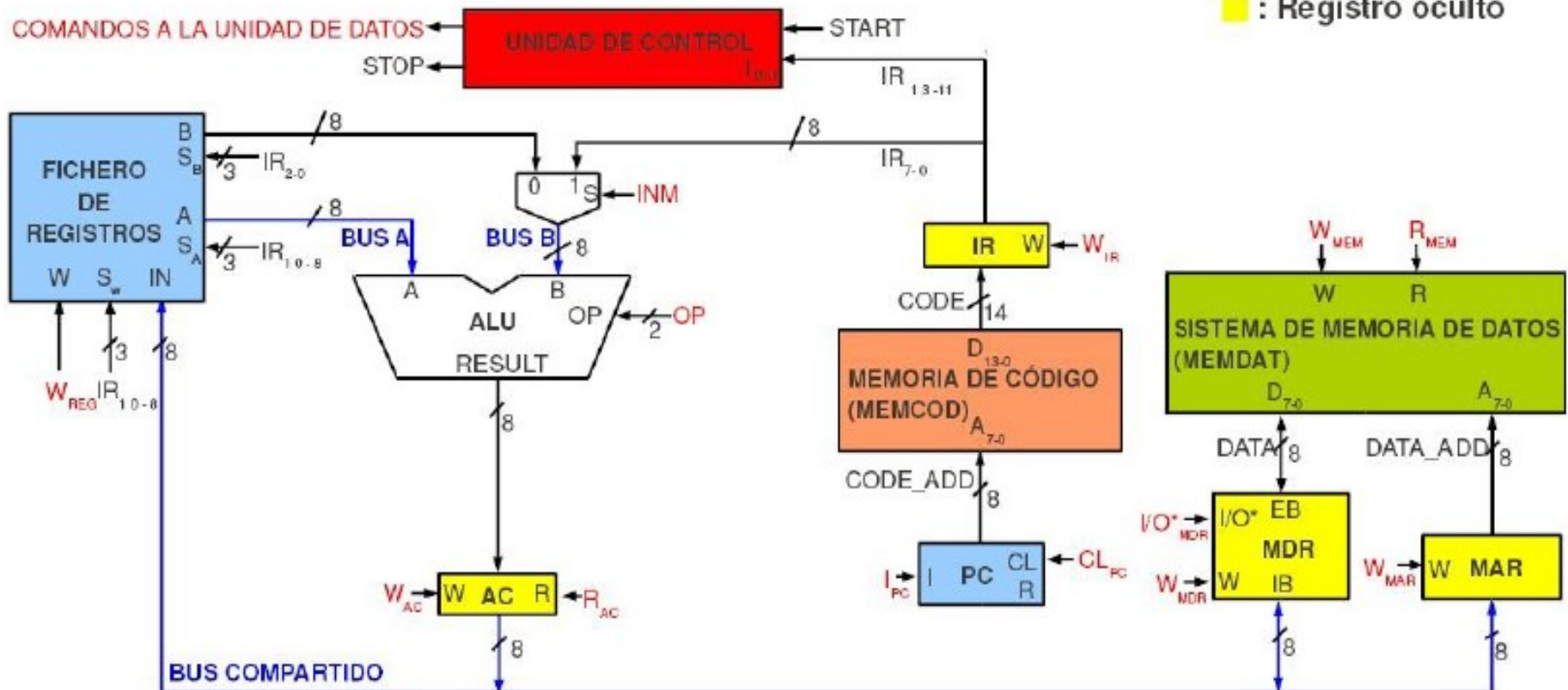
- Si el salto es condicional, se deberá comprobar si la condición es **cierta o falsa**. Si es cierta, se produce el salto, si es falsa éste no se produce (se continúa con la ejecución lineal)
- Las condiciones están representadas por banderines **N, Z, V, C** que proceden de la salida de la ALU y que se deben almacenar en un registro o conjunto de biestables denominado **registro de estado** o SR
- Permiten implementar “salta si el resultado es 0”, “salta si son distintos”, “salta si es mayor que”, “salta si....”



Arquitectura del CS2

Adaptación para salto condicional
(A completar por el alumno)

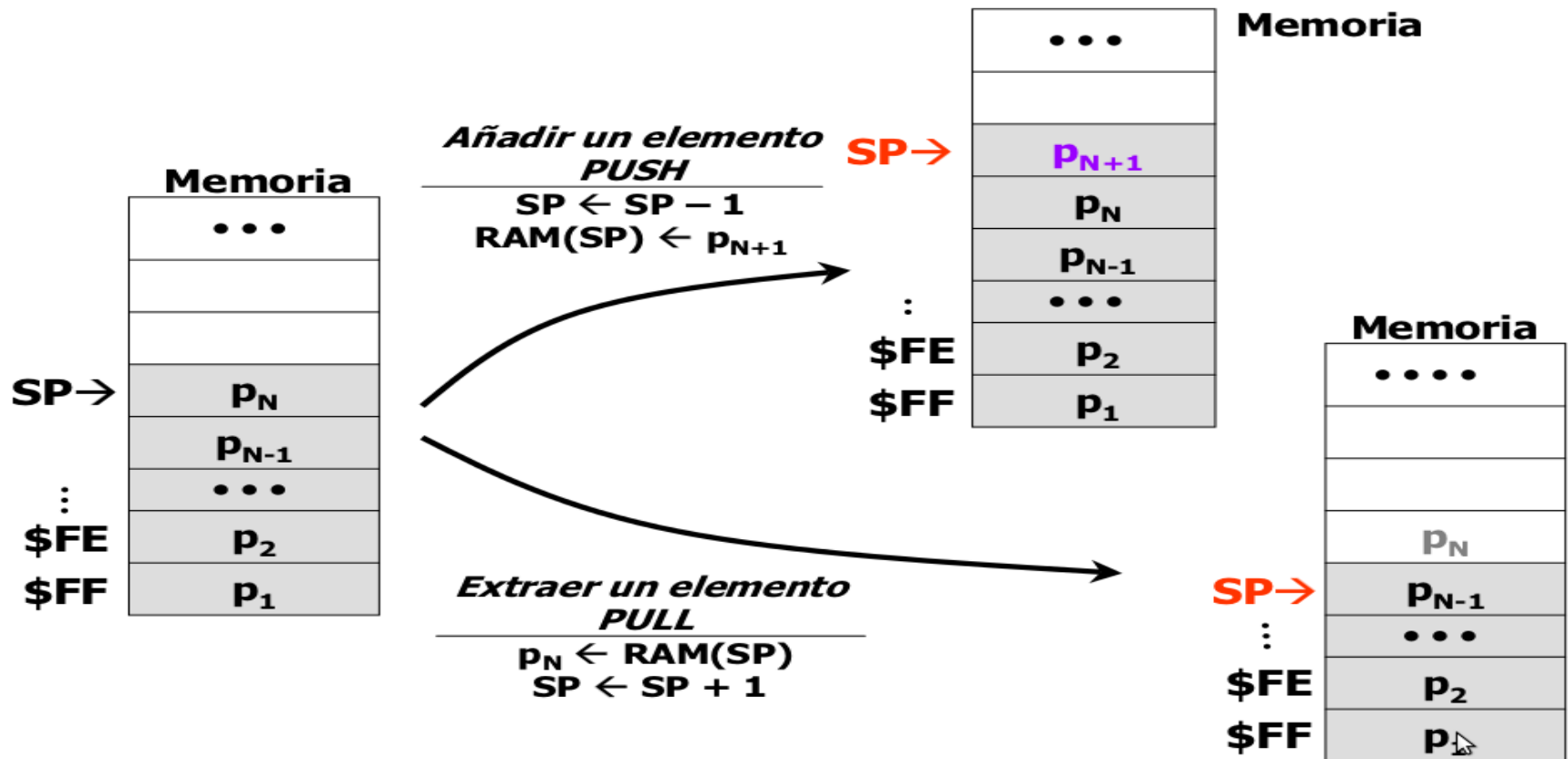
■ : Registro visible
■ : Registro oculto



Instrucciones de llamada y retorno de subrutinas

- La pila es una estructura de datos que reside en memoria
- El registro SP (Stack Pointer, puntero de pila) apunta a la cima de la pila
- Operaciones sobre la pila:
 - PUSH: añadir un elemento en la cima de la pila
 - PULL: extraer el elemento situado en la cima de la pila
- La pila permite los saltos y retornos de subrutinas: instrucciones CALL y RET

Instrucciones de llamada y retorno de subrutinas

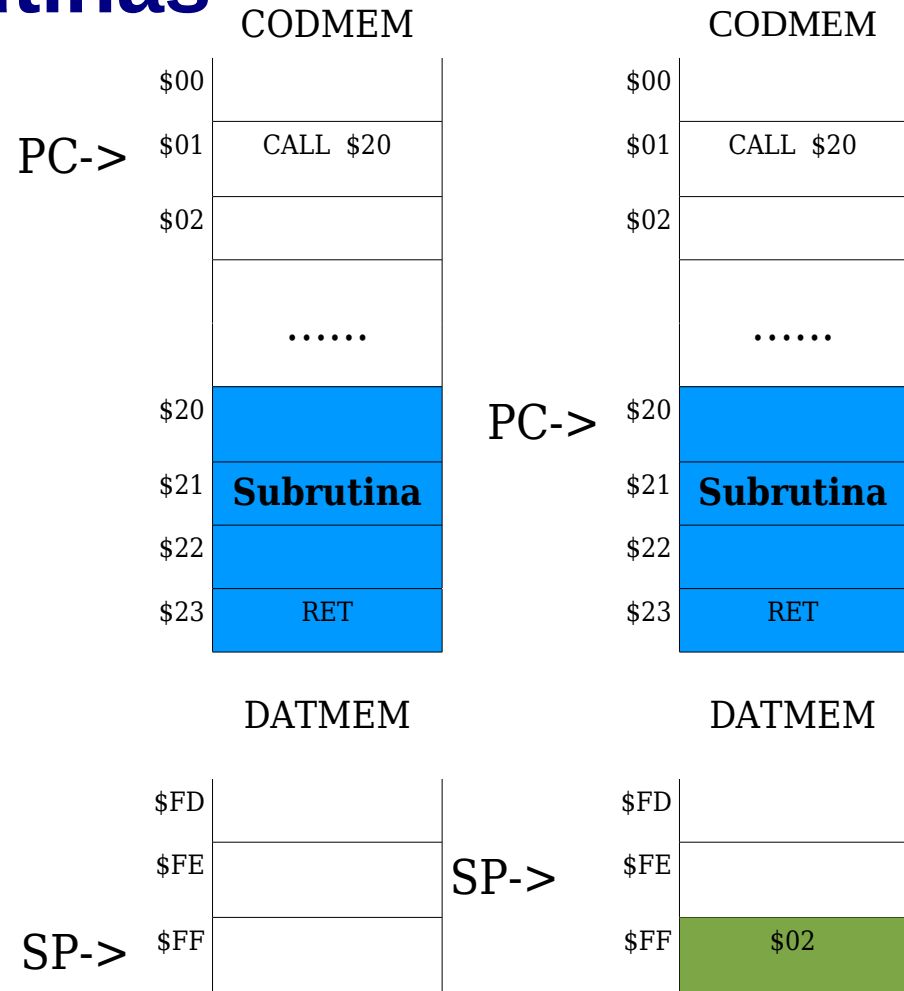


Instrucciones de llamada y retorno de subrutinas

- Una subrutina es una secuencia de instrucciones que realizan una cierta tarea a partir de unos datos de entrada, generando unos datos de salida
- Cuando un programa necesita realizar la tarea, hace una llamada a la subrutina, ésta se ejecuta y cuando termina, el programa se reanuda por donde se había quedado (retorno de subrutina).
- Una subrutina se puede llamar todas las veces que el programa lo necesite
- Una subrutina puede llamar a su vez a otra subrutina: ejecución anidada de subrutinas
- Una subrutina puede llamarse a sí misma: subrutinas recursivas

Instrucciones de llamada y retorno de subrutinas

- Situación inicial (a). El computador ejecuta instrucciones y llega a CALL \$20.
- La ejecución de la instrucción CALL \$20 hace que:
 - Se guarde en la pila el PC (que contiene \$03)
 - Se “salte” a la dirección \$20
- Observe el comportamiento del puntero de pila o registro SP (b).

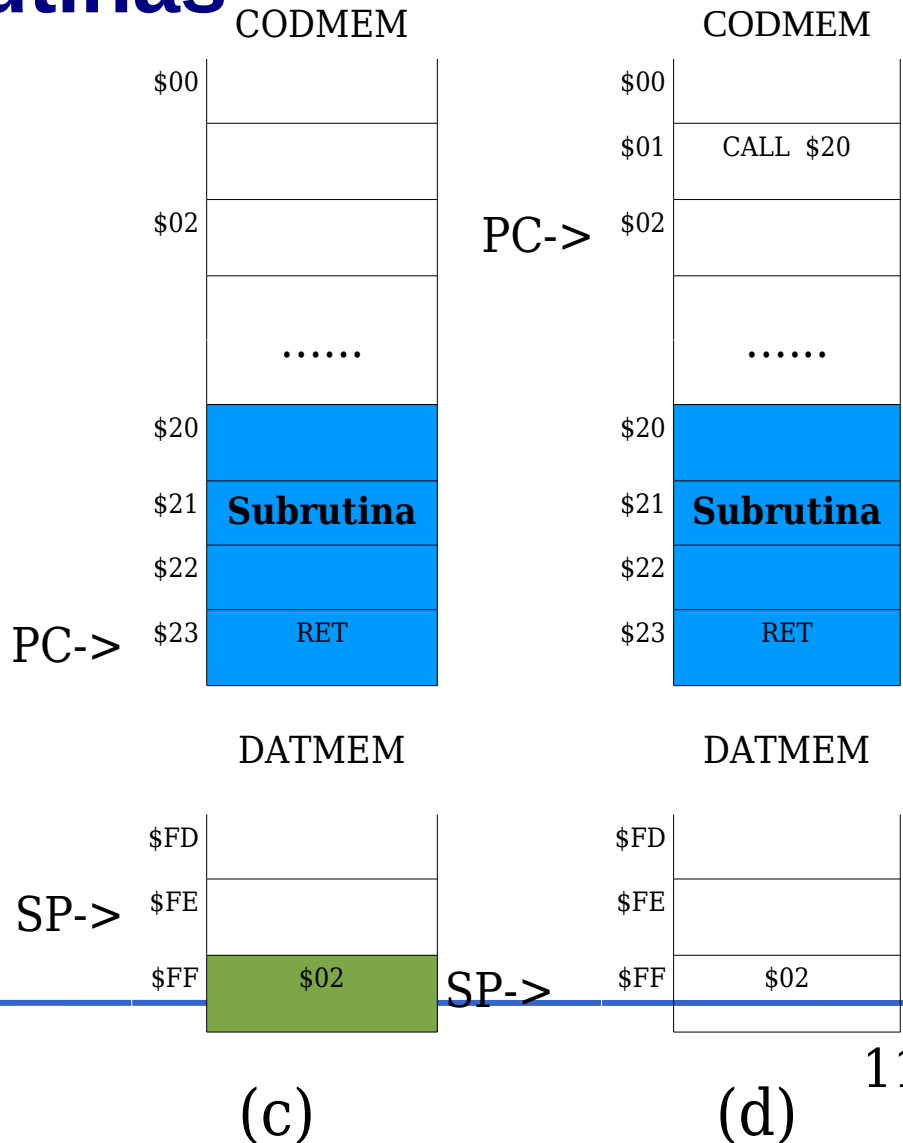


(a)

(b)

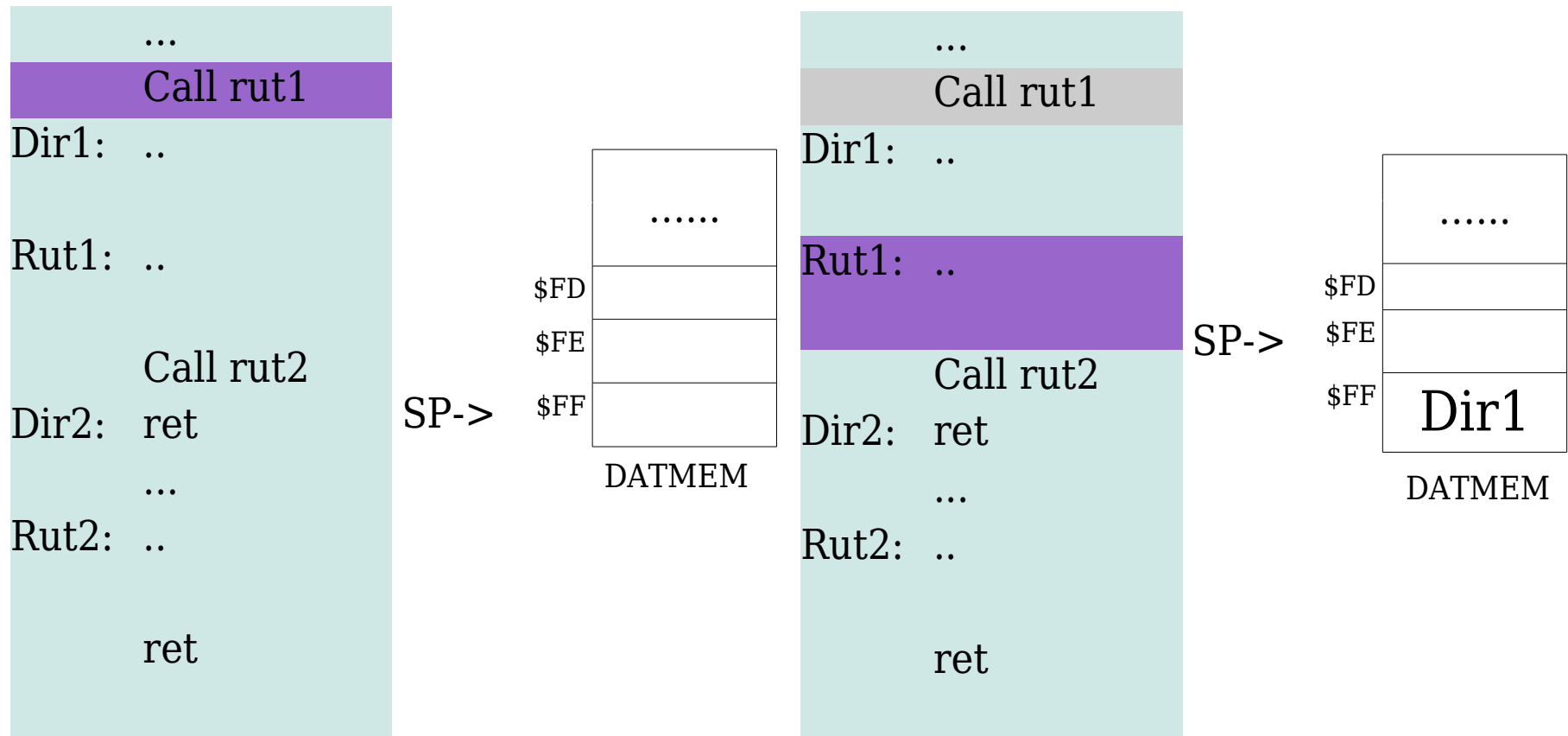
Instrucciones de llamada y retorno de subrutinas

- Se ejecutan las instrucciones de la subrutina hasta que se procede a ejecutar la última instrucción o RET (c)
- La ejecución de RET hace que se saque el dato que está en la cima de la pila y que se guarde sobre el PC (d)
- A continuación se sigue ejecutando instrucciones. En este caso a partir de la dirección \$03.
- Observe que \$03 sigue en la PILA pero ya no es accesible y en la próxima operación de PUSH será sobrescrito.



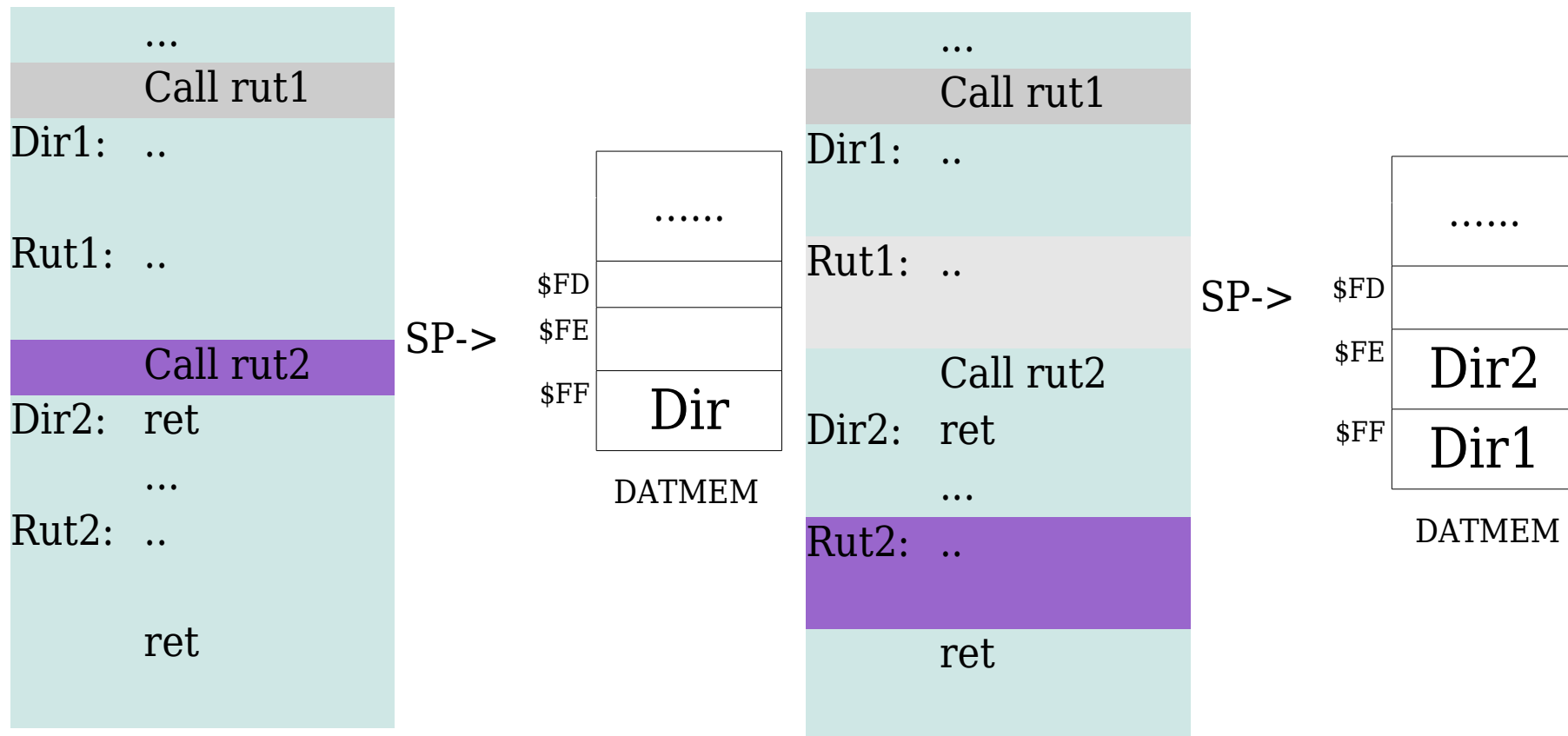
Instrucciones de llamada y retorno de subrutinas

- Anidamiento de subrutinas



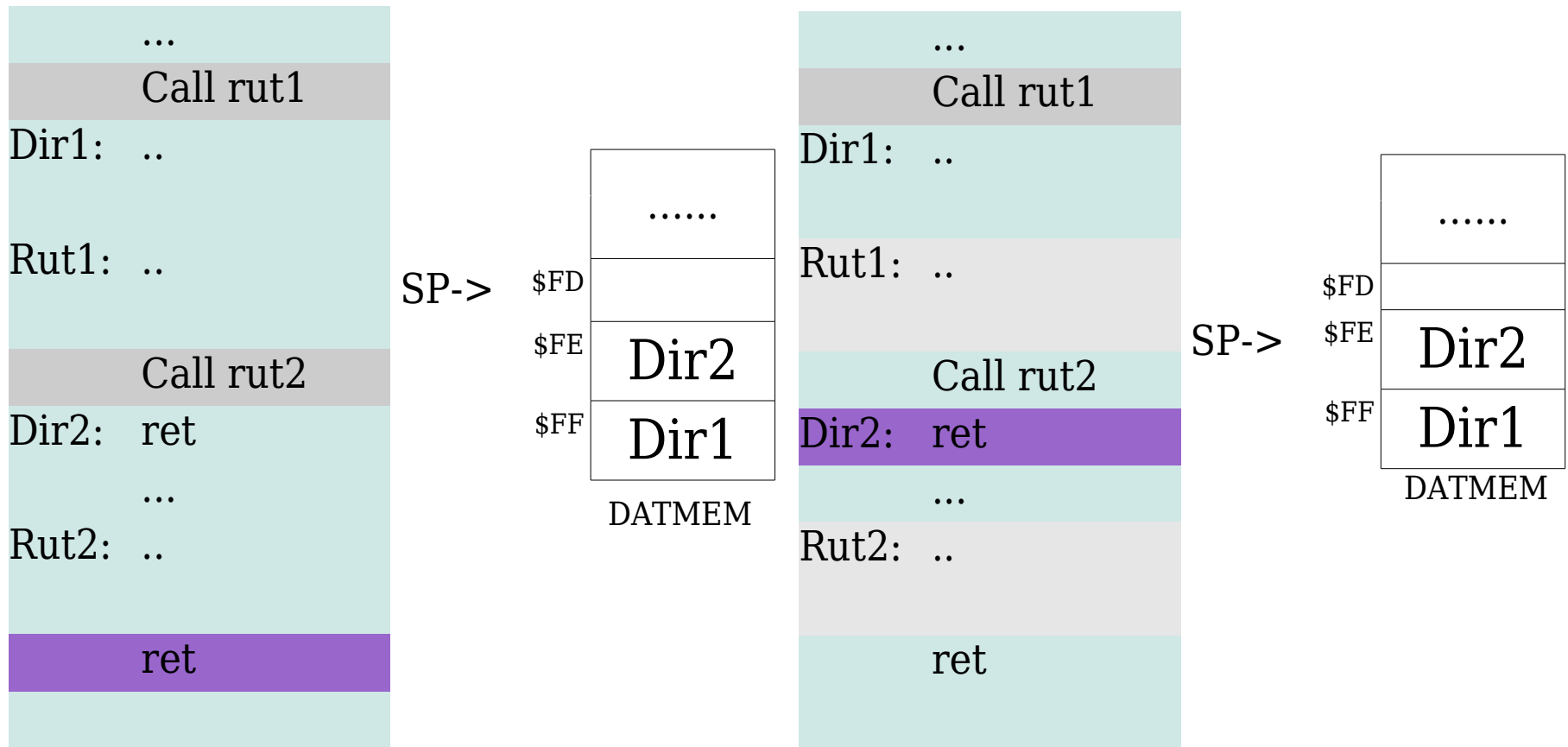
Instrucciones de llamada y retorno de subrutinas

- Anidamiento de subrutinas



Instrucciones de llamada y retorno de subrutinas

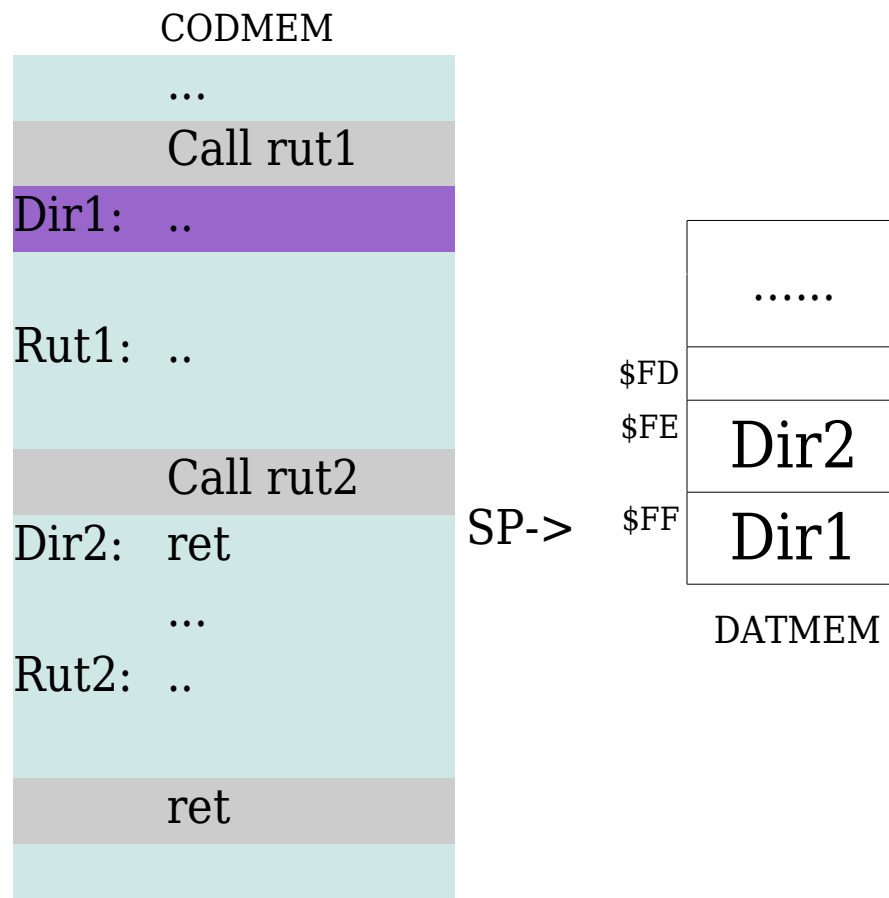
- Anidamiento de subrutinas



Instrucciones de llamada y retorno de subrutinas

Anidamiento de subrutinas

- Observe que el orden en el que se deben extraer los contenidos del PC de la PILA debe ser en sentido inverso a como éstos se introdujeron en ella. (LIFO)



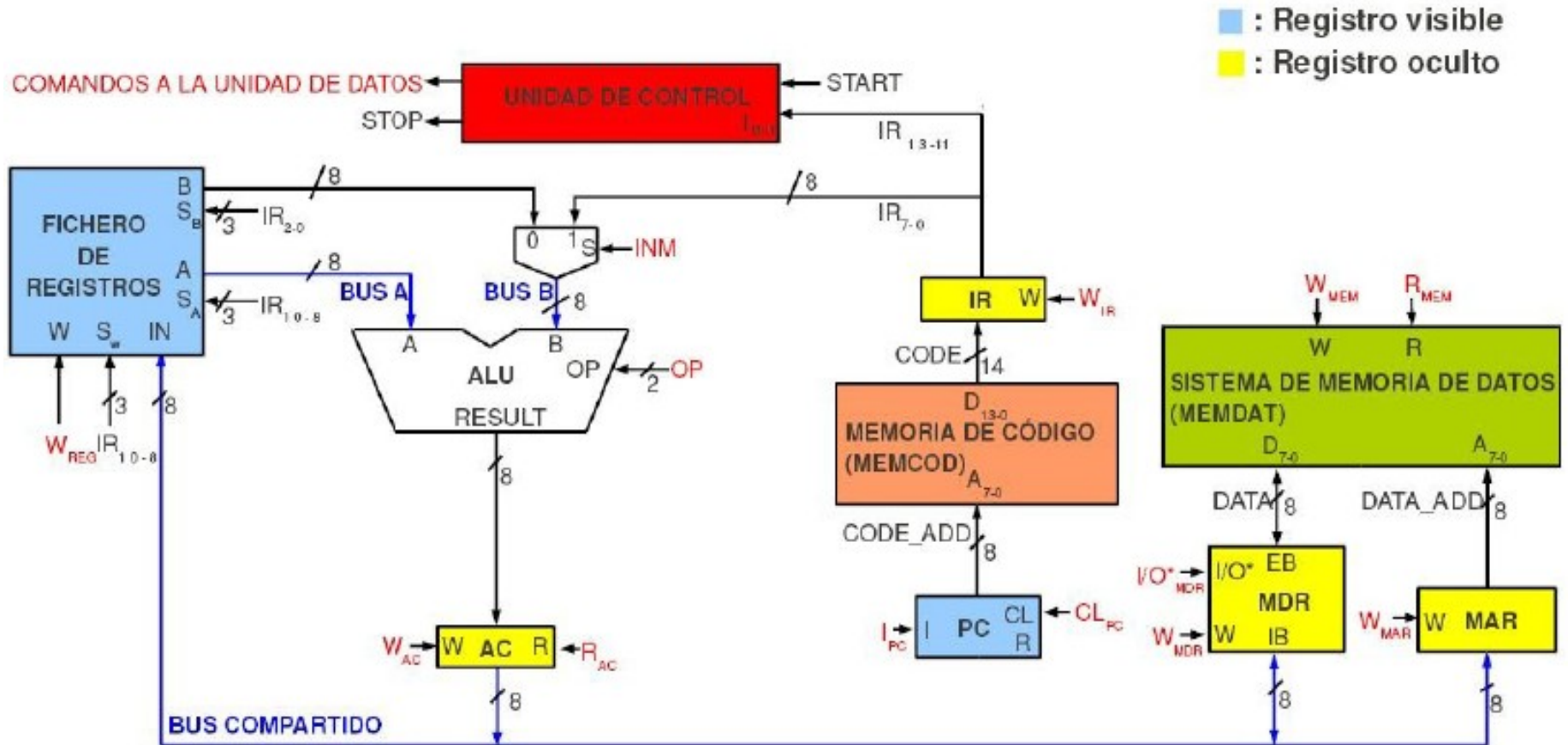
Instrucciones de llamada y retorno de subrutinas

- Formato de las instrucciones CALL y RET



Arquitectura del CS2

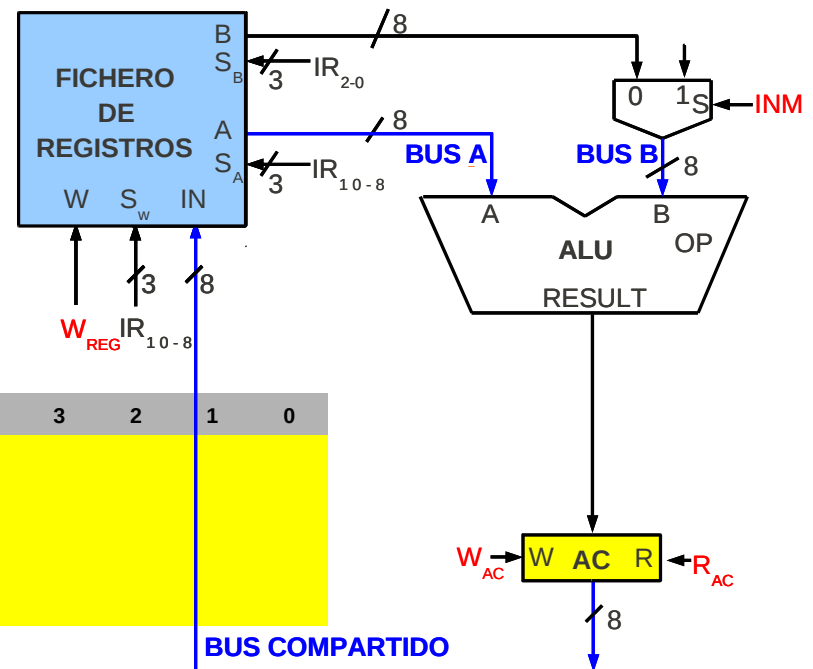
Adaptación para llamadas y retorno de subrutinas
(A completar por el alumno)



Rotación de un registro

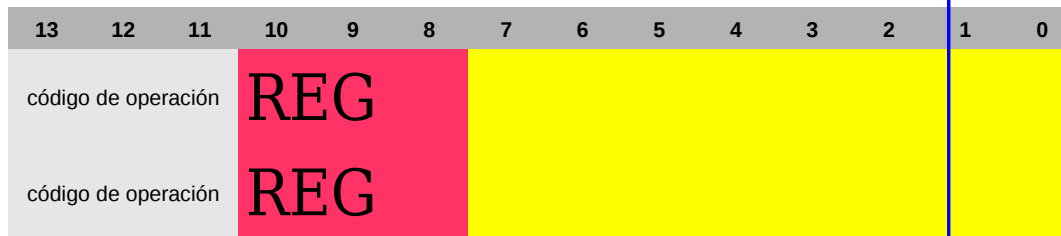
(a completar por el alumno)

- Una solución pasaría por añadirle señales de rotación a cada uno de los registros=> 16 señales de control. Se complica con el acarreo (C).
- La solución más simple pasaría por tener una ALU con capacidad de desplazar a derecha e izquierda de manera combinacional (esta es la solución que implementa el CS2010).



formato

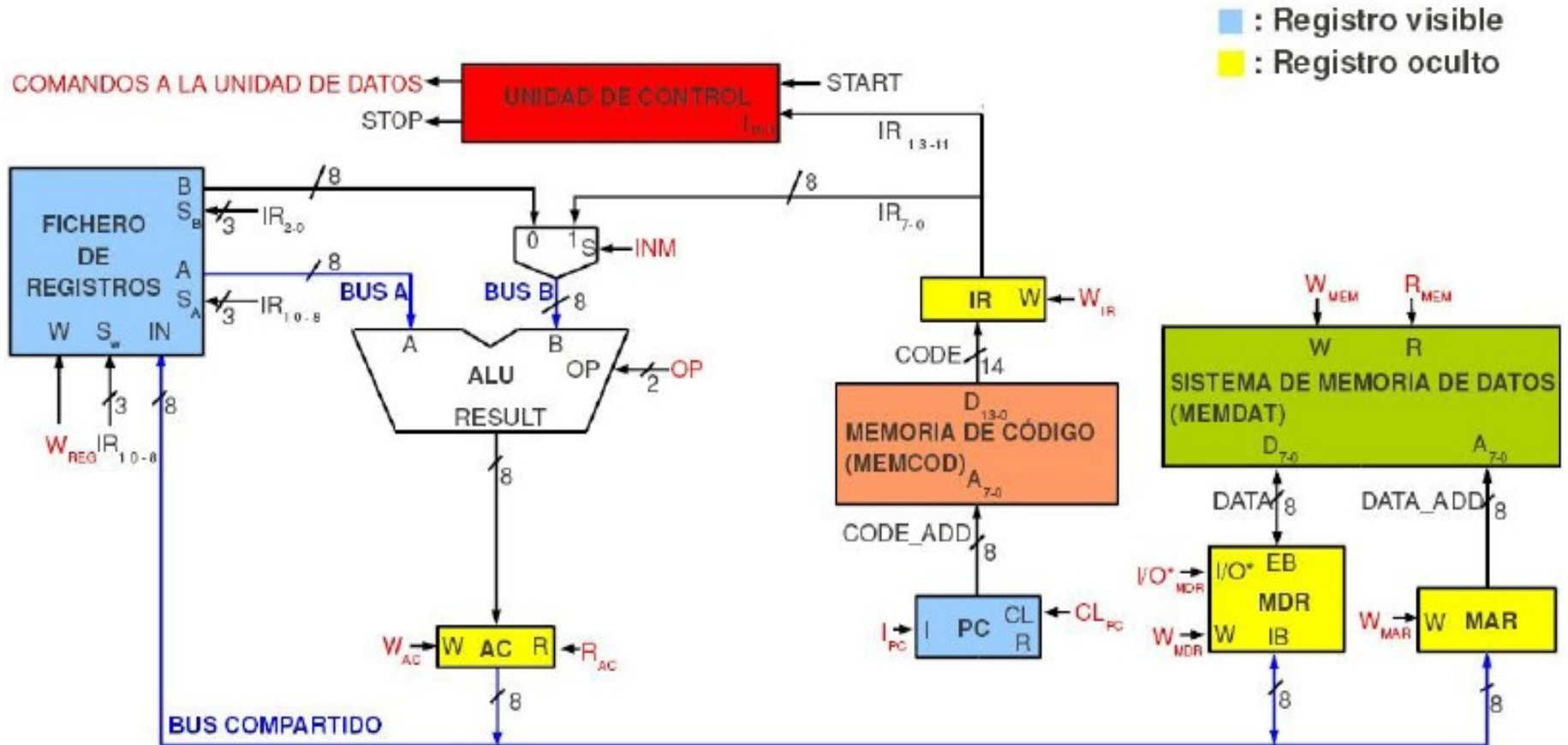
ror
rol



Incrementar el número de instrucciones

- Al aumentar el número de instrucciones de un computador, se necesita incrementar el número de bits de la instrucción. (código de operación)
- Pasar de 8 instrucciones del CS2 a 32 posibles en el CS2010, exige instrucciones de 16 bits.
- Se deben **modificar los anchos de algunos buses**.
- Se incrementa la complejidad de la unidad de datos
- Aumenta el número de líneas de control y la funcionalidad de la unidad de control.

Incrementar el número de instrucciones (A completar)



Instrucciones con modo de direccionamiento inmediato (A completar)

