

Boletín 3: El computador simple

Última modificación: 04/05/18

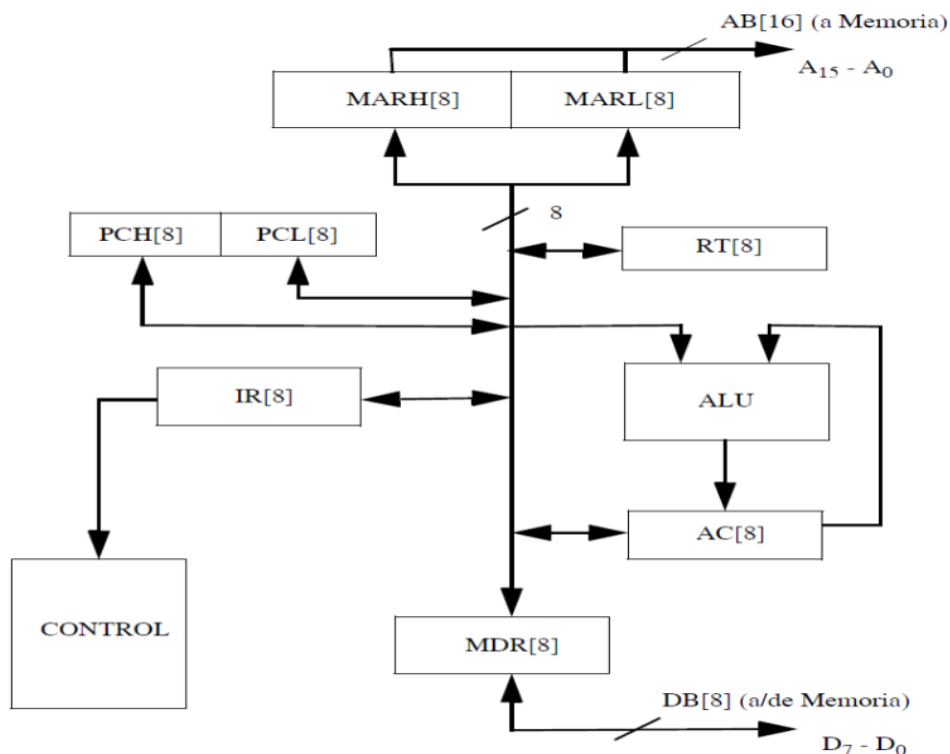
Problema 1

La siguiente figura muestra la Unidad de Procesado de Datos de un microprocesador con arquitectura von Neumann. Como puede observarse, el bus de datos (D7-D0) es de 8 bits y el de direcciones (A15-A0) de 16 bits. Los registros visibles son los siguientes:

- PC (Program Counter): es de 16 bits y está formado por la concatenación de dos registros de 8 bits (PCH y PCL).
- AC (Acumulador): es de 8 bits.

Además, la unidad de procesado dispone de los siguientes registros ocultos:

- MAR (Memory Address Register): de 16 bits, está formado por la concatenación de dos registros de 8 bits: MARH y MARL.
- MDR (Memory Data Register): de 8 bits, su finalidad es servir como registro intermedio entre el procesador y la memoria externa. Todo dato que entre o salga del procesador deberá ser almacenado previamente en MDR.
- IR (Instruction Register): es de 8 bits.
- RT (Registro Temporal): es de 8 bits y es utilizado para el cálculo de operaciones intermedias.



Boletín 3: El computador simple

Las posiciones de memoria tienen un ancho de 8 bits, y cada instrucción ocupa tres posiciones consecutivas: la primera indica el código de operación, la segunda contiene los 8 bits más significativos del operando y la tercera los 8 bits menos significativos. Consideremos por ejemplo la instrucción LDA. Dicha instrucción escribe el contenido de la posición de memoria indicada por el operando en AC. La instrucción 'LDA \$B043' ocuparía 3 posiciones: en la primera se encontraría el código de operación correspondiente a LDA, en la segunda habría escrito \$B0 y en la tercera \$43. Indique la secuencia de microoperaciones necesarias para realizar dicha instrucción y diga cuáles corresponden al ciclo de FETCH y cuáles al de EXECUTE.

Problema 2

Se desea cambiar el juego de instrucciones del CS1 por el que se muestra en la siguiente tabla:

Código de operación ($IR_{7,6}$)	Instrucción
00	$Rd \leftarrow Rd + 2 Rf$
10	$Rd \leftarrow Rd - 2 Rf$
01	$Rd \leftarrow 2 Rf$
11	$Rd \leftarrow 0$

Dibuje las cartas ASM de datos y control que permitan implementar estas nuevas instrucciones.

Problema 3

El CS1 es un sistema digital que busca la ejecución de instrucciones de forma automática alternando dos fases o ciclos: Búsqueda (Fetch) o Ejecución (Execute), hasta que se ejecuta la instrucción STOP. Indique:

- ¿Cuántos ciclos de reloj se necesitan para ejecutar (incluyendo la búsqueda) cada instrucción del CS1?
- La latencia (*latency*) de una instrucción se define como el tiempo que tarda la CPU en buscarla y ejecutarla. ¿Cuál es la latencia promedio del CS1?
- El rendimiento o producción (*throughput*) es el número de instrucciones que se pueden ejecutar expresadas en ciclos de reloj. Dicho de otra forma, cuántas instrucciones ejecuta el CS1 en un ciclo de reloj. Calcule el rendimiento del CS1.
- Debido a que las operaciones a realizar con los datos almacenados en el fichero de registros y la búsqueda de instrucciones están separados en el *datapath* (unidad de datos), se podría pensar en la posibilidad de solapar en el tiempo ambos ciclos, de modo que mientras que una instrucción se está ejecutando, se puede proceder a buscar la siguiente. Así se consigue mayor *throughput*. Realice las modificaciones de la carta ASM del CS1 para que esto sea posible.
- Finalmente calcule la latencia y el rendimiento de esta versión mejorada del CS1.

Nota: Puede ampliar los conceptos de eficiencia y rendimiento en wikipedia: http://en.wikipedia.org/wiki/Instruction_pipeline

Problema 4

Se desea añadir dos nuevas instrucciones del CS2. Una de ellas es la instrucción NOP, que no realiza ninguna operación, mientras que la otra permite la suma de un operando inmediato con un registro Rd.

Código de operación (IR_{15-11})	Instrucción
xxxxx	NOP
xxxxx	$Rd \leftarrow Rd + \text{operando}$

Dibuje las cartas ASM de datos y control modificadas para que se realicen estas instrucciones

Problema 5

Los modos indirectos del CS2 incluyen sólo a los registros R7, R6 que se denominan Z e Y respectivamente. No sería complejo extender este concepto al resto de los registros. Para no tener que usar nuevos caracteres del alfabeto para denominar a los registros que serán utilizados en el modo indirecto, suponga que el formato usado fuera (Rm), es decir, la instrucción LD Rn,Y pasaría a ser LD Rn,(R6) y, en general, LD Rn,(Rm), donde $m=0..7$. Indique cuáles serían los cambios, si los hubiera, que habría que llevar a cabo en el CS2 para extender el modo indirecto a todos los registros del fichero.

Problema 6

Se desean añadir al CS3 dos instrucciones que permiten la inserción (PUSH Ri) y extracción (POP Ri) de datos en la pila. Asigne un código de operación adecuado e indique sus microoperaciones (véase la Figura 4 y la Tabla 3).

Problema 7

Se desea cambiar el modo de direccionamiento de las instrucciones ST y LD del CS3 para que usen el modo indirecto con postincremento. La operación es idéntica a la del modo indirecto normal salvo que, tras realizarse el acceso a memoria, el registro base se incrementa en una unidad.

- Indique qué cambios habría que realizar a la unidad de datos del sistema para que pudiera implementarse dicho modo (véase Figura 4).
- Descomponga las nuevas instrucciones en microoperaciones.

Problema 8

Diseñe una subrutina para el CS3 que calcule la división de dos números sin signo. El dividendo está almacenado en el registro R0 y el divisor en R1, que se supondrá distinto de cero. El cociente debe quedar guardado en R2 y el resto en R3.

Problema 9

Se tiene una tabla de 10 números con signo almacenados a partir de la dirección \$34 de la memoria. Diseñe una subrutina para el CS3 que calcule su suma.

Boletín 3: El computador simple

Problema 10

En el CS3 la condición a testear en los saltos condicionales se codifica en los bits 8, 9 y 10 de la instrucción (véase Figura 6). Hay cuatro códigos de condición no utilizados a los que se desea asociar las siguientes condiciones de salto:

I_{10}	I_9	I_8	CONDICIÓN	mnémico(s) de la condición	notas
1	0	0	NOT Z	NE	Será cierta justo tras realizar la resta A-B si y solo si $A \neq B$
1	0	1	NOT C	CC, SH	Será cierta justo tras realizar la resta A-B si y solo si $A \geq B$ asumiendo notación base 2 sin signo
1	1	0	NOT V	VC	Será cierta si y solo si el dato recién calculado es representable asumiendo notación con complemento a 2
1	1	1	N nexor V	GE	Será cierta justo tras realizar la resta A-B si y solo si $A \geq B$ en notación con complemento a 2

Indique los cambios que habría que realizar sobre las cartas ASM del CS3 para implementar las nuevas condiciones de salto.

Problema 11

Se desea añadir las instrucciones STD y LDD al CS3 para que usen direccionamiento indirecto de registro con desplazamiento. En este modo de direccionamiento el dato se encuentra en una posición de memoria cuya dirección se obtiene sumando el contenido de un registro (registro base -Y o Z-) y una constante codificada en la propia instrucción (desplazamiento). Su sintaxis ensamblador será:

LDD Rd, Y+q o LDD Rd, Z+q

STD Y+q, Rf o STD Z+q, Rf

A modo de ejemplo, al ejecutarse las siguientes instrucciones el registro Y se escribiría con el valor 23 y el registro R1 se escribiría con el contenido de la posición de memoria 40.

LDI R6, 23

LDD R1, Y+17

El desplazamiento se codificará en los bits no usados del formato A (véase Figura 6) en notación complemento a dos.

- Indique como habría que modificar la unidad de datos de la implementación propuesta para hacer esto posible (véase Figura 4).
- Indique como habría que modificar las cartas ASM de datos y control.

Problema 12

Obtenga las secuencias de micro-operaciones para las siguientes instrucciones del CS3

- a) CALL dir
- b) RET
- c) JMP dir
- d) BRxx dir
- e) CP
- f) CPI

Problema 13

Para las instrucciones aritméticas del CS2, que se incorporan al CS3, ¿sería necesario modificar su secuencia de micro-operaciones? Si es que sí, indique cómo.

Problema 14

Se desea introducir la instrucción ADC Rd,Rf al CS3. Esta instrucción genera el siguiente resultado $Rd \leftarrow Rd + Rf + C$, donde C es el valor del acarreo del registro de estado. Haga las modificaciones necesarias de la unidad de datos para que la solución se ejecute en el mismo número de ciclos de reloj que la instrucción ADD Rd,Rf.

Problema 15 dic-2015

Se desean implementar dos nuevas instrucciones en el CS3:

ADDPUSH Ri,Rj que suma los registros Ri y Rj y guarda el resultado en la pila.

ADDPOP Ri,Rj que extrae dos datos de la pila, los guarda en los registros Ri y Rj y posteriormente los suma quedando el resultado en Ri.

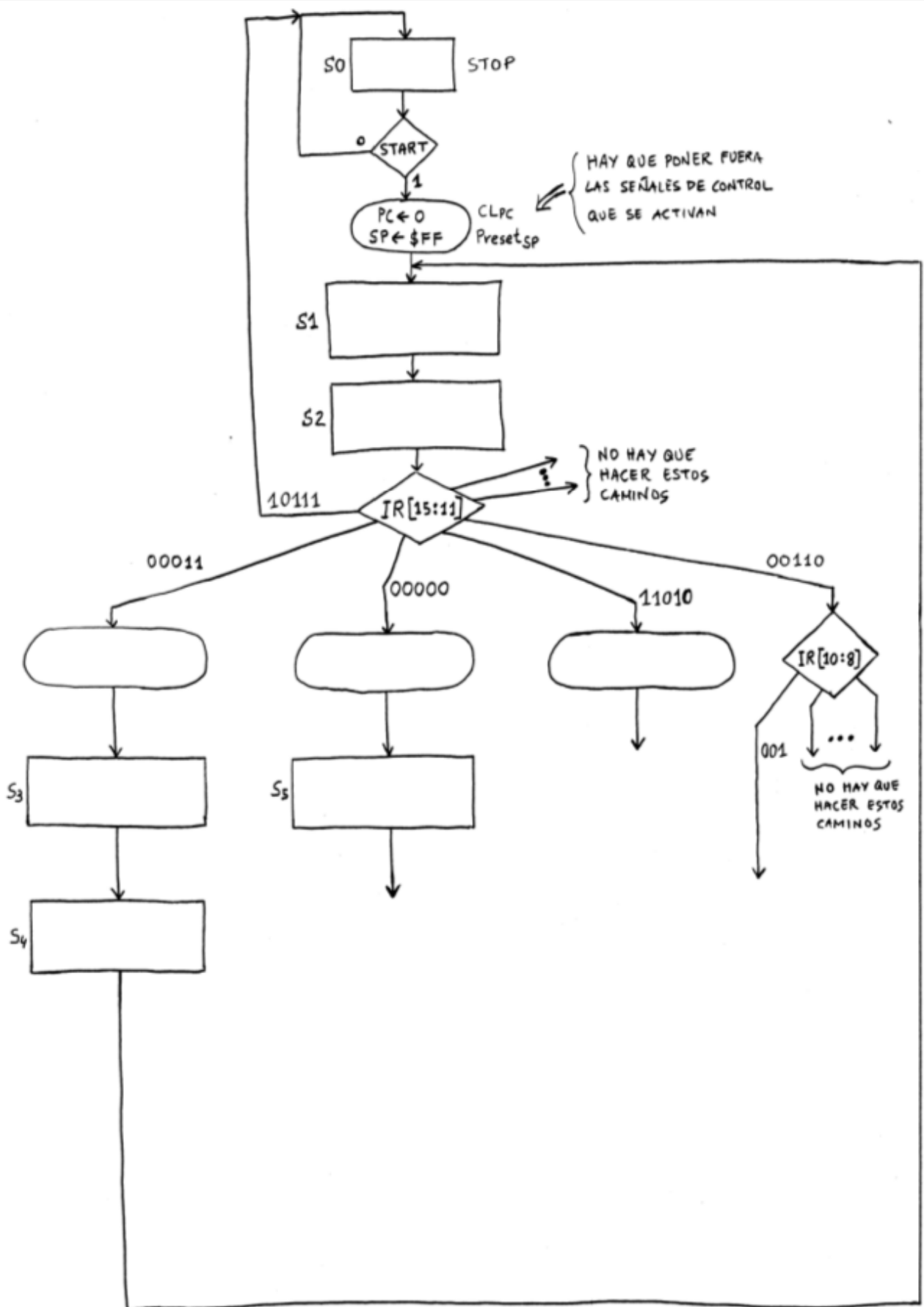
Las instrucciones utilizan el formato A, pero los registros no siempre son fuente y destino. En el caso de ADDPUSH ambos son fuente y en el de ADDPOP ambos son destino. Se pide:

- (a) Asignar un código de operación válido para ambas instrucciones.
- (b) La instrucción ADDPOP requiere una pequeña modificación en el hardware de la unidad de datos. Explique y dibuje en que consiste dicha modificación.
- (c) Desarrolle las micro-operaciones de ambas instrucciones. Se deberá considerar tanto el ciclo de búsqueda como el de ejecución y poner tanto las transferencias entre registros como las señales a activar por la unidad de control.

Problema 16 Sep-2015

En la figura siguiente se muestra la carta ASM de la unidad de control del CS3. Debe completarla, siguiendo las instrucciones que aparecen en la figura. Añada los bloques ASM que faltan, rellene los que están vacíos, y deje sin hacer aquello marcado como "NO HAY QUE HACER". No olvide indicar las señales de control.

Boletín 3: El computador simple



Problema 17 jun-2015

Se desea añadir al **CS3** una nueva instrucción, **MOVW**, cuya sintaxis es la siguiente: **MOVW R_{d+1}:R_d, R_{f+1}:R_f** siendo **d,f ∈ {0, 2, 4, 6}**.

- Asigne a la nueva instrucción el código de operación y el formato de código máquina que considere oportunos.
- Describa las modificaciones necesarias en la Unidad de Datos para su implementación.
- Obtenga la secuencia de microoperaciones de MOVW en la nueva unidad de datos (transferencias RT y señales a activar).

Problema 18 dic-2015

Suponga que la situación actual de CODMEM y DATMEM del CS3 es la mostrada en las tablas que aparecen más abajo, que la Unidad de Control del CS3 está en el estado S0 y que el resto de registros y elementos de memoria no especificados se encuentran a valor desconocido.

Explique razonadamente cuáles serán los valores finales de los 8 registros de propósito general, del AC, del IR, del PC, del SP, del MAR y de la memoria tras activar la entrada START de la unidad de control y esperar a que ésta genere la salida STOP.

CODMEM																
Posi- ción	Contenido															
0	1	1	1	1	1	1	1	0	0	0	0	0	0	1	1	1
1	0	0	0	0	1	1	1	1	0	1	0	0	1	1	1	0
2	1	1	0	1	0	1	1	0	0	0	0	0	0	0	1	1
3	0	0	1	1	0	0	0	0	0	0	0	0	0	1	1	1
4	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0
5	0	0	0	1	1	1	1	0	0	0	0	0	0	1	0	1
6	0	0	1	1	1	1	0	1	0	0	0	0	0	0	0	1
7	1	0	1	1	1	1	0	1	0	0	0	0	0	1	1	0

DATMEM	
Posi- ción	Conte- nido
0	8
1	7
2	6
3	5
4	4
5	3
6	2
7	1

Boletín 3: El computador simple

Problema 19 sep-2015

Responda a las siguientes cuestiones:

- Explique qué características del hardware del CS3 nos permiten implementar las instrucciones de salto.
- ¿En qué se basa la unidad de control del CS3 para evaluar si se cumple la condición en las instrucciones de salto condicional?
- ¿Qué diferencia existe entre las instrucciones CALL y JMP?
- Tomando como punto de partida la arquitectura del CS3, determine a qué afecta o qué cambios debe hacer para los siguientes casos (tomados por separado):
 - a. Ampliar la memoria de programa (más palabras, pero de igual tamaño)
 - b. Ampliar la longitud de las instrucciones (seguirán siendo de una palabra pero con más bits)
 - c. Ampliar la memoria de datos (más palabras, pero de igual tamaño)
 - d. Ampliar la longitud de los datos
 - e. Permitir que el AC sea un operando de la ALU
 - f. Permitir instrucciones con más de una palabra de código máquina

Problema 20 jun-2015

Se desea añadir una nueva instrucción, EXG que tiene el siguiente formato: EXG Rd,Rf y que intercambia el contenido de los registros Rd y Rf, esto es: $Rd \leftrightarrow Rf$.

- a) Asigne un formato de instrucción y un código de operación concreto.
- b) Describa las modificaciones que se deben realizar en la unidad de datos para poderla implementar, teniendo en cuenta que deben ser las menos posibles y no implicar cambios en el funcionamiento de las instrucciones ya existentes.
- c) Describa la secuencia de microoperaciones necesarias para llevarlas a cabo (tanto transferencias RT como señales de control).

ANEXO

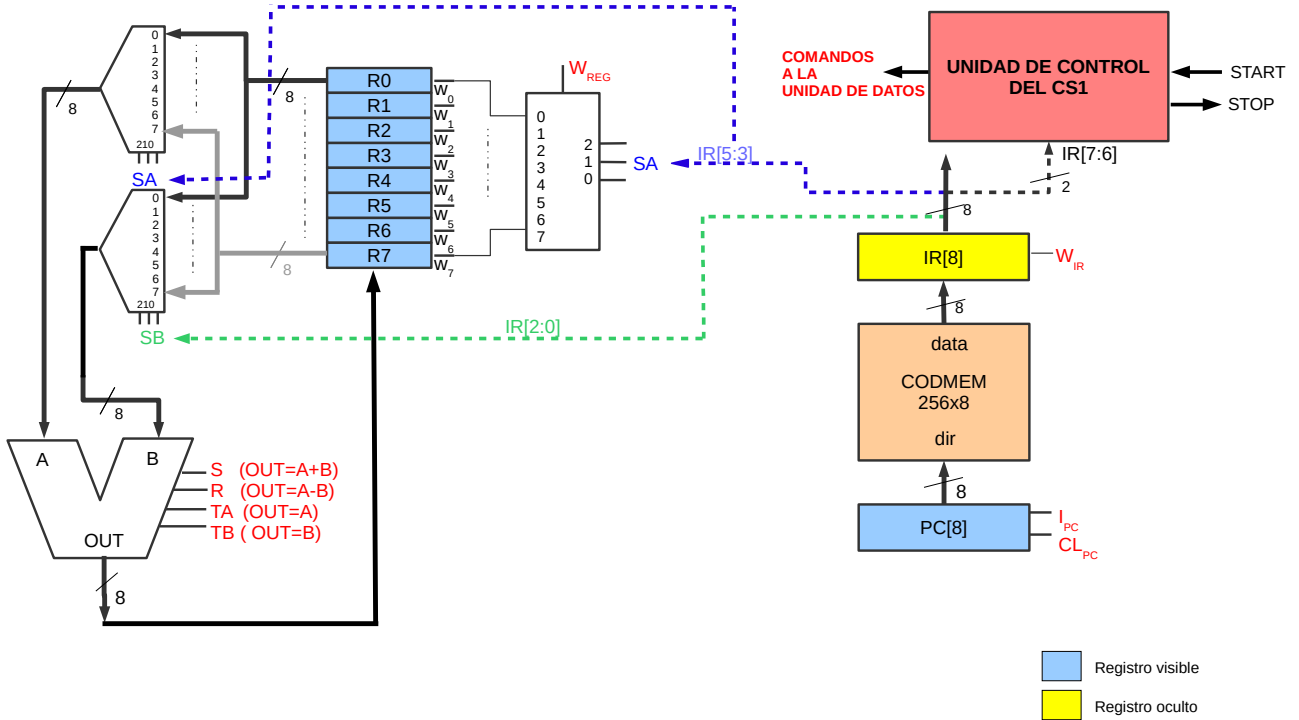


Figura 1: Unidad de datos del CS1

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

Tabla 1: Juego de instrucciones del CS1

Boletín 3: El computador simple

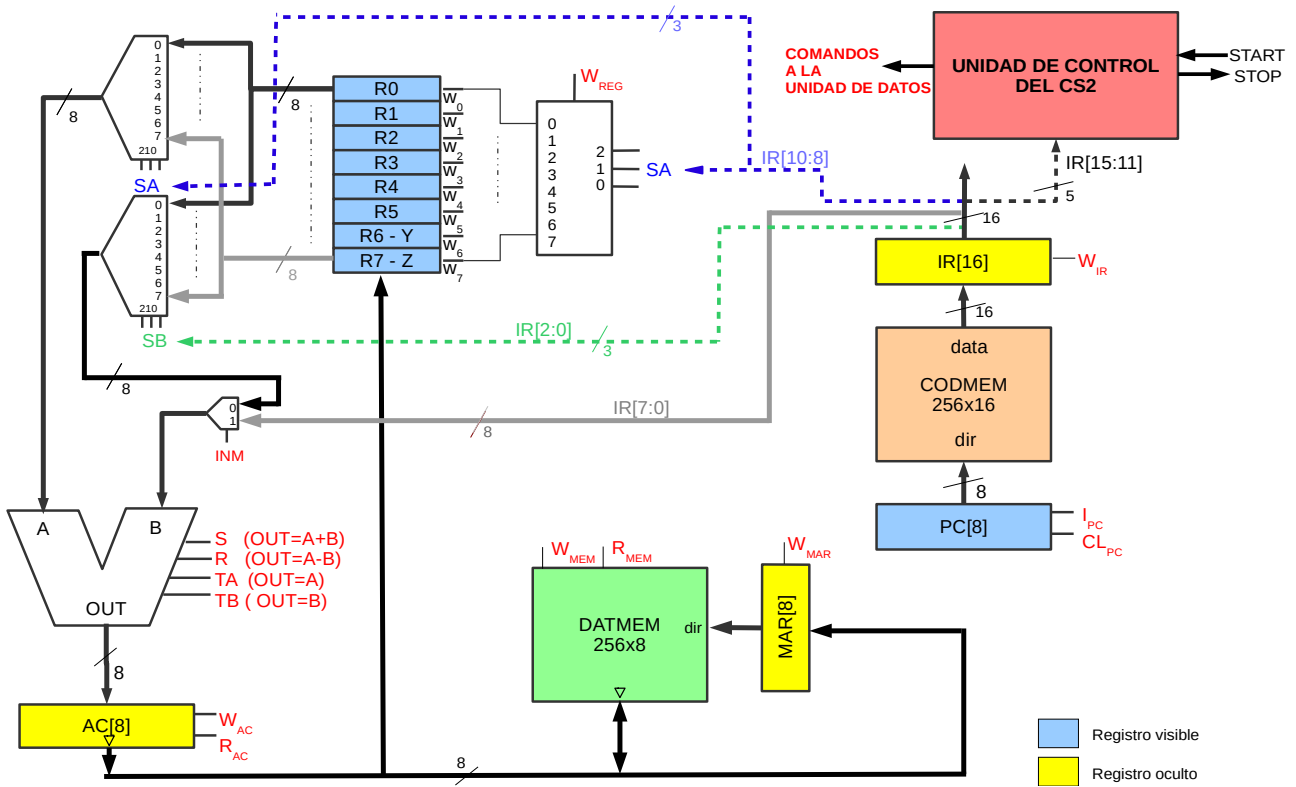


Figura 2: Unidad de datos del CS2

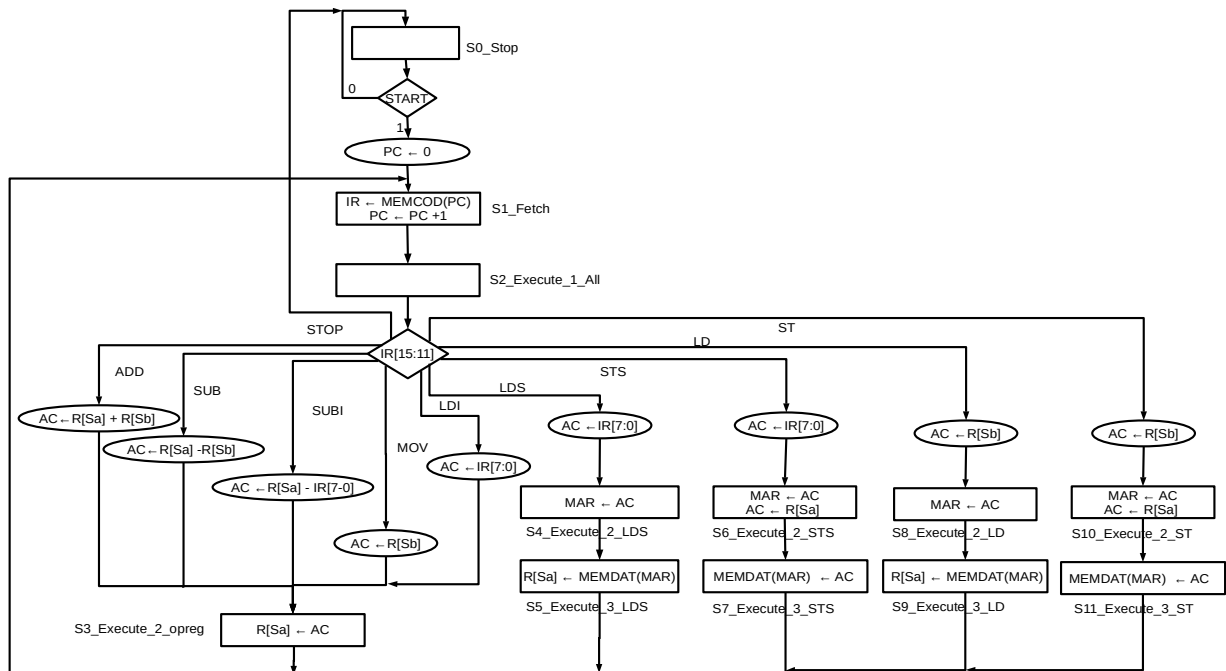


Figura 3: Carta ASM de datos y control del CS2

Boletín 3: El computador simple

CO	SINTAXIS	FUNCIÓN
00000	ST Y o Z,Rf	MEMDAT(Y o Z)←Rf
00001	LD Rd,Y o Z	Rd←MEMDAT(Y o Z)
00010	STS dir,Rf	MEMDAT(dir)←Rf
00011	LDS Rd,dir	Rd←MEMDAT(dir)
11111	LDI Rd,dato	Rd ← dato
01000	ADD Rd,Rf	Rd←Rd+Rf
01010	SUB Rd,Rf	Rd←Rd-Rf
11010	SUBI Rd,dato	Rd←Rd-dato
01111	MOV Rd,Rf	Rd←Rf
10111	STOP	NOP

Tabla 2: Juego de instrucciones del CS2

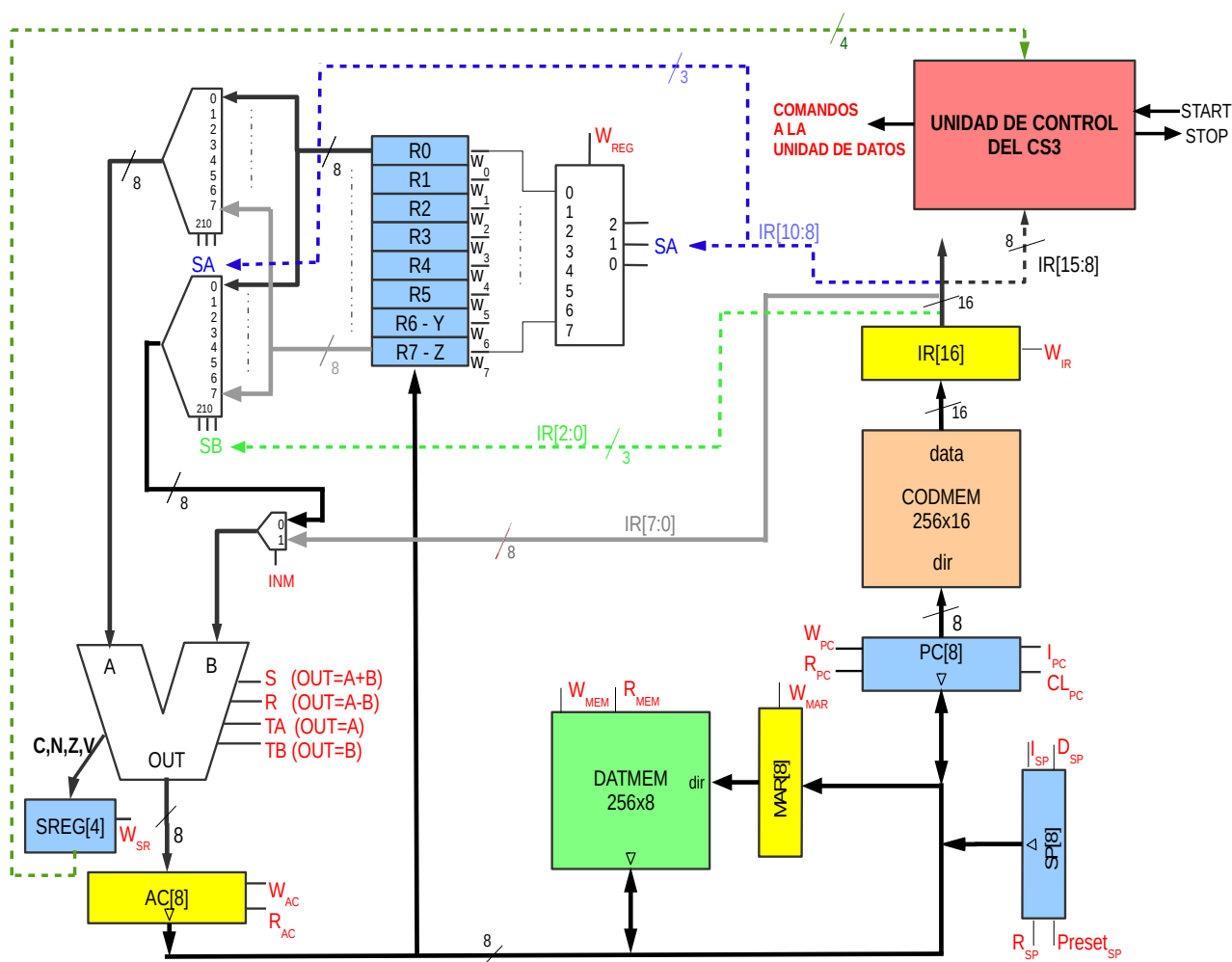


Figura 4: Unidad de datos del CS3

Boletín 3: El computador simple



Figura 5: Formato de las instrucciones del CS3

Bits del código de operación					NEMÓNICO	FORMATO	TIPO	SINTAXIS	EFECTO ¹	VCNZ ²
15	14	13	12	11						
0	0	0	0	0	ST	A	memoria	ST YoZ,Rfuente	MEM[YoZ]←Rfuente	----
0	0	0	0	1	LD	A	memoria	LD Rdestino,YoZ	Rdestino←MEM[YoZ]	----
0	0	0	1	0	STS	B	memoria	STS dirección,Rfuente	MEM[dirección]←Rfuente	----
0	0	0	1	1	LDS	B	memoria	LDS Rdestino,dirección	Rdestino←MEM[dirección]	----
0	0	1	0	0	CALL	C	salto	CALL dirección	MEM[SP]←PC,SP←SP-1, PC←dirección	----
0	0	1	0	1	RET	-	salto	RET	PC←MEM[SP+1],SP←SP+1	----
0	0	1	1	0	BRxx	C	salto	BRxx dirección	xx: PC←dirección	----
0	0	1	1	1	JMP	C	salto	JMP dirección	PC←dirección	----
0	1	0	0	0	ADD	A	aritmético/lógica	ADD Rdestino,Rfuente	Rdestino←Rdestino+Rfuente	****
0	1	0	1	0	SUB	A	aritmético/lógica	SUB Rdestino,Rfuente	Rdestino←Rdestino-Rfuente	****
0	1	0	1	1	CP	A	estado	CP Rdestino,Rfuente	Rdestino-Rfuente	****
0	1	1	1	1	MOV	A	movimiento de datos	MOV Rdestino,Rfuente	Rdestino←Rfuente	----
1	0	1	1	1	STOP	-	especial	STOP	lleva el procesador a espera	----
1	1	0	1	0	SUBI	B	aritmético/lógica	SUBI Rdestino,dato	Rdestino←Rdestino-dato	****
1	1	0	1	1	CPI	B	estado	CPI Rdestino,dato	Rdestino-dato	****
1	1	1	1	1	LDI	B	movimiento de datos	LDI Rdestino,dato	Rdestino←dato	----

¹ (sin tener en cuenta el registro de estado y el incremento del PC)

² El caracter '-' denota "no modificado", "*" denota "modificado de forma definida"

Tabla 3 Juego de instrucciones del CS3